PERANCANGAN SISTEM KEAMANAN JARINGAN BERBASIS SOFTWARE DEFINED NETWORK (SDN) MENGGUNAKAN INTRUSION DETECTION AND PREVENTION SYSTEM (Studi Kasus: Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Sains dan Teknologi

Oleh :

JONI PADJRI
11455101839

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2021

LEMBAR PERSETUJUAN

TUGAS AKHIR

Oleh :

JONI PADJRI

11455101839

Tugas Akhir ini telah diperiksa dan disetujui sebagai Laporan Tugas Akhir Program Studi Teknik Elektro di Pekanbaru, pada tanggal 06 Agustus 2021

Pembimbing

Oktaf Brillian Kharisma, S.T.,M.T.

NIP. 198410122015031003

Ketua Program Studi

Dr. Zulfatri Aini, S.T.,M.T.

NIP. 197210212006042001

LEMBAR PENGESAHAN

PERANCANGAN SISTEM KEAMANAN JARINGAN BERBASIS SOFTWARE DEFINED NETWORK (SDN) MENGGUNAKAN INTRUSION DETECTION AND PREVENTION SYSTEM (Studi Kasus: Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau)

TUGAS AKHIR

Oleh :

JONI PADJRI
11455101839

Telah dipertahankan di depan Seminar Proposal Tugas Akhir
salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau
di Pekanbaru, pada tanggal 06 Agustus 2021

Pekanbaru, 06 Agustus 2021

MENGESAHKAN,

Ketua Prodi Teknik Elektro

Digitally signed by Zulfatri Aini
Tanggal: 2021.08.25 13:01:40 WIB

Dr. Zulfatri Aini, S.T., M.T
NIP. 197210212006042001

DEWAN PENGUJI :

Ketua : Dr. Harris Simaremare, ST, MT — Digitally signed by Harris Simaremare Date: 2021-08-25 10:09:07:00

Sekretaris : Oktaf Brillian Kharisma, S.T, M.T. — Digitally signed by Oktaf Brillian Kharisma Tanggal: 2021.08.15 14:28:29 WIB

Anggota I : Ewi Ismaredah, S.Kom, M.Kom.

Anggota II : Abdillah, S.Si, MIT. — Digitally signed by Abdillah Tanggal: 23-08-2021 17:36:43

LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL

Tugas Akhir yang tidak diterbitkan ini terdaftar dan tersedia di Perpustakaan Universitas Islam Negeri Sultan Syarif Kasim Riau adalah terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada penulis. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau ringkasan hanya dapat dilakukan seizin penulis dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya.

Penggandaan atau penerbitan sebagian atau seluruh Tugas Akhir ini harus memperoleh izin dari Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau. Perpustakaan yang meminjamkan Tugas Akhir ini untuk anggotanya diharapkan untuk mengisi nama, tanda peminjaman dan tanggal pinjam.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa di dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan oleh saya maupun orang lain untuk kepentingan lain, dan sepanjang pengetahuan saya juga tidak memuat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali disebutkan dalam referensi dan di dalam daftar pustaka.

Saya bersedia menerima sanksi jika pernyataan ini tidak sesuai dengan yang sebenarnya.

Pekanbaru, 06 Agustus 2021

Yang Membuat Pernyataan,

Joni Padjri

11455101839

LEMBAR PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

أَلَمْ نَشْرَحْ لَكَ صَدْرَكَ ۝١ وَوَضَعْنَا عَنكَ وِزْرَكَ ۝٢ الَّذِي أَنقَضَ
ظَهْرَكَ ۝٣ وَرَفَعْنَا لَكَ ذِكْرَكَ ۝٤ فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ۝٥ إِنَّ مَعَ
الْعُسْرِ يُسْرًا ۝٦ فَإِذَا فَرَغْتَ فَانصَبْ ۝٧ وَإِلَىٰ رَبِّكَ فَارْغَب ۝٨

Bukankah Kami telah melapangkan untukmu dadamu? Dan Kami telah menghilangkan darimu bebanmu? Yang memberatkan punggungmu. Dan Kami tinggikan bagimu sebutanmu. Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan yang lain). Dan hanya kepada Tuhanmulah hendaknya kamu berharap.

(Q.S. Al – Insyirah :1-8)

Ya Allah ... dengan izin dan ridho-Mu atas segala kerendahan dan ketulusan hati, kupersembahkan buah goresan tangan ini sebagai tanda hormat dan baktiku untuk Ibunda Yuliarni, Ayahanda Zainuddin Caniago beserta Adik-adikku beserta keluarga yang selalu dan tak pernah letih menyayangiku, tak pernah lupa mendo'akan ku dan tak pernah bosan memberiku support baik materi maupun fikiran.

Tugas Akhir ini khusus kupersembahkan kepada :

1. Ayahanda Zainuddin Caniago dan Ibunda Yuliarni terima kasih atas kasih sayang, pengorbanan, ketulusan dan keiklasan serta do'anya yang tak pernah putus.
2. Untuk Adik-adikku yang telah memberikan dukungannya.
3. Kawan-kawan dekat seperjuangan Siswanto, Rio Susanto, Muhammad Dicky, Al Hafis, Gema, Fikri, Tengku Muhammad Iqbal dan rekan-rekan lainnya baik Senior maupun Junior yang tidak bisa dituliskan satu persatu. Terimaksihya telah banyak membantu, sukses untuk kita semua...Amin.

PERANCANGAN SISTEM KEAMANAN JARINGAN BERBASIS SOFTWARE DEFINED NETWORK (SDN) MENGGUNAKAN INTRUSION DETECTION AND PREVENTION SYSTEM
(STUDI KASUS: PUSAT TEKNOLOGI INFORMASI PANGKALAN DATA (PTIPD) UIN SUSKA RIAU)

JONI PADJRI
11455101839

Tanggal Sidang : 06 Agustus 2021

Program Studi Teknik Elektro
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas KM 15 No.155 Pekanbaru

ABSTRAK

Pertumbuhan jaringan komputer bergerak dengan sangat cepat seiring dengan pertumbuhan penggunanya ditambah banyaknya penggunaan perangkat jaringan yang masih melakukan konfigurasi manual menjadi kendala bagi *network administrator*. Dengan adanya jaringan *Software Defined Network* (SDN), *network* administrator dapat membentuk lalu lintas jaringan melalui sebuah *central console* pada kontroler, sehingga tidak perlu mengkonfigurasi masing-masing perangkat jaringan yang terdapat pada topologi. Pada Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau pengelolaan dan *konfigurasi* jaringan masih berbasis *Konvesional*, dengan hasil rata-rata *Quality of Service* (QoS) dari pengujian jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau dilakukan sebanyak 3 kali pengujian yaitu: *Troughput* 5.029 bps, *Packet Loss* 0,89 %, dan *Delay* 0,00115 ms. Pada jaringan berbasis *Software Defined Network* (SDN), dengan hasil rata-rata *Quality of Service* (QoS) yaitu *Troughput* 6.804 bps, *Packet Loss* 0,34 %, dan *Delay* 0,00111ms. *Controller* merupakan pusat dari seluruh jaringan komputer berkonsepkan *Software Defined Network* (SDN) yang bertanggung jawab atas keluar masuknya komunikasi data berupa penjadwalan dan trafik pada jaringan tersebut, sehingga *controller* lebih cenderung mendapatkan serangan *Distributed Denial of Service* (DDoS). Pada pengujian serangan *Distributed Denial of Service* (DDoS) menggunakan botnet di Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau yang dilakukan 3 kali pengujian, dengan hasil rata-rata *Quality of Service* (QoS) yaitu: *Troughput* 8,743 bps, *Packet Loss* 29,43%, dan *Delay* 0,0767 ms. Sistem keamanan Jaringan yang mampu mengidentifikasi terhadap serangan-serangan yang ada yaitu *Intrusion Detection and Prevention System* (IDPS) menggunakan *snort*. Pada pengujian *Intrusion Detection and Prevention System* (IDPS) yang dilakukan 3 kali pengujian, dengan hasil rata-rata *Quality of Service* (QoS) yaitu: yaitu t*hroughput* 414,67 bps, *delay* 0,00771 ms, *packet loss* 4,14%.

Kata Kunci: *Software Defined Network* (SDN), *Intrusion Detection and Prevention System* (IDPS), *snort*, *botnet*, *Quality of Service* (QoS)

***DESIGN OF NETWORK SECURITY SYSTEM BASED ON SOFTWARE DEFINED NETWORK (SDN) USING INTRUSION DETECTION AND PREVENTION SYSTEM CASE STUDY: DATABASE INFORMATION TECHNOLOGY CENTER* (PTIPD) UIN SUSKA RIAU)**

JONI PADJRI
1145510 1839

Date of Session : 06 *August* 2021

Departement of Electrical Engineering
Faculty of Science and Technology
State Islamic University of Sultan Syarif Kasim Riau
Soebrantas St. No.155 Pekanbaru - Indonesia

ABSTRACT

The growth of computer networks moves very quickly along with the growth of users plus the number of use of network devices that still perform manual configuration is an obstacle for network administrators. With the Software Defined Network (SDN) network, network administrators can shape network traffic through a central console on the controller, so there is no need to configure each network device contained in the topology. At the Information Technology Center and Database (PTIPD) of UIN Suska Riau, the management and configuration of the network is still conventional based, with the average Quality of Service (QoS) results from testing the network of the Information Technology Center and Database (PTIPD) UIN Suska Riau carried out as many as 3 testing times are: throughput 5.029 bps, Packet Loss 0.89%, and Delay 0.00115 ms. On a Software Defined Network (SDN) based network, the average Quality of Service (QoS) results are 6.804 bps throughput, 0.34% packet loss, and 0.00111 ms delay. The controller is the center of the entire computer network with a Software Defined Network (SDN) concept which is responsible for the entry and exit of data communications in the form of scheduling and traffic on the network, so that the controller is more likely to get Distributed Denial of Service (DDoS) attacks. In testing the Distributed Denial of Service (DDoS) attack using a botnet at the Information Technology and Database Center (PTIPD) UIN Suska Riau which was carried out 3 times, with the average Quality of Service (QoS) results, namely: throughput 8,743 bps, Packet Loss 29.43%, and Delay 0.0767 ms. A network security system that is able to identify existing attacks is the Intrusion Detection and Prevention System (IDPS) using snort. In the Intrusion Detection and Prevention System (IDPS) test, 3 tests were carried out, with the average Quality of Service (QoS) results, namely: throughput 414.67 bps, delay 0.00771 ms, packet loss 4.14%.

Keywords: *Software Defined Network (SDN), Intrusion Detection and Prevention System (IDPS) , snort ,botnet, Quality of Service (QoS)*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb

Puji syukur saya ucapkan atas kehadirat Allah SWT yang telah memberikan seluruh rahmat dan karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan laporan Tugas Akhir ini sebagai salah satu syarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar sarjana di program studi Teknik Elektro Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.

Sholawat dan salam selalu disampaikan kepada Baginda Nabi Muhammad SAW yang telah membawa umatnya dari zaman kegelapan menuju zaman yang terang benderang seperti saat sekarang ini.

Penulisan Tugas Akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Sains dan Teknologi. Atas berkat rahmat dan ridho Allah SWT penulis dapat menyelesaikan Tugas Akhir ini dengan judul "PERANCANGAN SISTEM KEAMANAN JARINGAN BERBASIS *SOFTWARE DEFINED NETWORK* (SDN) MENGGUNAKAN *INTRUSION DETECTION AND PREVENTION SYSTEM* (Studi Kasus: Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau).

Dalam penyelesaian Laporan Tugas Akhir ini penulis mendapat bimbingan, bantuan, dan dukungan yang sangat berarti dari berbagai pihak. Untuk itu penulis mengucapkan banyak terima kasih kepada :

1. Allah SWT yang telah memberi rahmat dan hidayah-Nya.
2. Orang tua yang sangat disayangi, atas doa, nasehat, cinta dan kasih sayang, yang telah memberikan motivasi dan semangat dari awal kehidupan hingga saat sekarang ini.
3. Ibu Dr. Zulfatri Aini, S.T., M.T. Selaku Ketua Program Studi Teknik Elektro periode 2021/2025 UIN SUSKA RIAU yang telah berkenan memberikan kesempatan melalui kebijakan administrasi hingga akhirnya penulis mampu menyelesaikan laporan Tugas Akhir penulis.
4. Bapak Dr. Harris Simaremare, ST, MT. Selaku Ketua Sidang yang senantiasa memberikan motivasi, dorongan dan solusi hingga akhirnya penulis mampu menyelesaikan laporan Tugas Akhir penulis.

5. Bapak Oktaf Brillian Kharisma, S.T, M.T. selaku Pembimbing yang telah meluangkan waktu, ilmu dan pikiran serta sabar dalam membimbing sehingga penulis dapat menyelesaikan laporan Tugas Akhir penulis.
6. Ibu Ewi Ismaredah, S.Kom, M.Kom. selaku Penguji I yang telah meluangkan waktu, ilmu dan pikiran dalam mengkoreksi hasil penelitian sehingga penulis dapat menyelesaikan laporan Tugas Akhir penulis.
7. Bapak Abdillah, S.Si, MIT. selaku Penguji II yang telah meluangkan waktu, ilmu dan pikiran dalam mengkoreksi hasil penelitian sehingga penulis dapat menyelesaikan laporan Tugas Akhir penulis.
8. Dan yang terakhir untuk sahabat-sahabat terbaik, Siswanto, Rio Susanto, Muhammad Dicky, Al Hafis, Gema, Fikri, Tengku Muhammad Iqbal dan teman-teman yang lainnya yang tidak bisa disebutkan satu per satu, yang telah memberikan dukungan yang sangat berarti selama ini.

Pada penulisan Tugas Akhir ini masih jauh dari kesempurnaan, karena kesempurnaan hanyalah milik Allah SWT dan kekurangan datang dari penulis sendiri. Dalam hal ini penulis menyadari bahwa Tugas Akhir ini masih memiliki kekurangan dan jauh dari kesempurnaan karena keterbatasan ilmu, pengalaman dan pengetahuan penulis dalam proses pembuatan Tugas Akhir ini, maka dari itu untuk penyempurnaan Tugas Akhir ini penulis mengharapkan kritikan dan saran kepada semua pihak yang sifatnya membangun.

Pekanbaru, Agustus 2021

Penulis,

Joni Padjri
11455101839

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Gambar **Halaman**

DAFTAR TABEL

Tabel **Halaman**

DAFTAR RUMUS

Rumus **Halaman**

DAFTAR SINGKATAN

= *Software Defined Network*

= *Intrusion Detection and Prevention System*

= *Open Network Operating System*

= *Robot Network*

= *Quality of Service*

= Pusat Teknologi Informasi dan Pangkalan Data

= *Internet Protocol*

= *Denial of Service*

= *Distributed Denial of Service*

DAFTAR GRAFIK

Lampiran **Halaman**

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan jaringan komputer bergerak dengan sangat cepat seiring dengan pertumbuhan penggunanya ditambah banyaknya penggunaan perangkat jaringan yang masih melakukan konfigurasi manual menjadi kendala bagi *network administrator*. Jaringan *Software Defined Network* telah membawa keuntungan yang sangat besar dalam dalam teknologi jaringan komputer. Jaringan *Software Defined Network* (SDN) merupakan teknologi jaringan dalam bidang arsitektur. Dengan adanya jaringan SDN, *network* administrator dapat membentuk lalu lintas jaringan melalui sebuah *central console* pada kontroler, sehingga tidak perlu mengkonfigurasi masing-masing perangkat jaringan yang terdapat pada topologi. Konsep utama dari teknologi SDN adalah untuk mendesain, membangun dan mengelola jaringan komputer secara terpusat dengan memisahkan secara fisik *control plane* dengan data *plane* [1].

Dalam jaringan *Software Defined Network* (SDN) yang memiliki sistem *controller* lebih mudah mendapatkan masalah karena *controller* dapat diprogram secara dinamis baik pada topologi maupun dalam pe-*routingan*. *Controller* juga merupakan pusat dari seluruh jaringan komputer berkonsepkan SDN yang bertanggung jawab atas keluar masuknya komunikasi data berupa penjadwalan dan trafik pada jaringan tersebut, sehingga c*ontroller* lebih cenderung mendapatkan serangan *Distributed Denial of Service* (DDoS). Namun. Isu keamanan jaringan adalah isu yang masih terbuka luas untuk diteliti pada jaringan SDN.

Penelitian terkait membahas tentang "Analisis Dampak Serangan DDoS Pada Jaringan *Openflow*." Hasil penelitiannya menunjukkan bahwa penggunaan CPU dan memory meningkat secara signifikan dalam waktu 30 detik. Dampak serangan DDoS dapat menganggu arus data dari user legal yang ada dalam jaringan untuk mengakses data tertentu karena layer kontrol dan pengiriman data sibuk melakukan processing data *dummy* (palsu) yang dikirimkan oleh *attacker*[2]. Penelitian terkait membahas tentang Dampak serangan DDoS pada *software based openflow switch* di perangkat HG553. Dalam penelitiannya menunjukkan UDP *Flooding* yang terjadi dalam jaringan *Software Defined Network* menyebabkan jaringan *Software Defined Network* sulit untuk di akses dikarenakan *Switch Opeflow* terlepas dari kontroller dan menyebabkan penggunaan

eoutine tinggi pada *Switch Openflow*[3]. Dari dua penelitian diatas dapat disimpulkan bahwa jaringan SDN rentan terhadap serangan *Distributed Denial of Service* (DDoS).

Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau merupakan salah satu unit pelaksanaan teknis yang menggunakan jaringan komputer. Menurut Bapak Indra Mulia Syafutra, S.T. sebagai staff yang mengelola jaringan komputer di Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau mengatakan bahwa Pusat Teknologi dan Informasi Pangkalan Data masih menggunakan jaringan konvensional yaitu masih melakukan konfigurasi secara manual atau satu persatu pada perangkat jaringan seperti *switch* dan *router* serta perangkat lunak jaringan lainnya didalam *infrastruktur* jaringan komputer Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau.

Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau menggunakan topologi *star* terdiri dari 3 lantai dan terdapat ruangan yang menggunakan jaringan komputer dengan jumlah keseluruhan 7 ruangan yaitu pada lantai 1 ada 3 ruangan dan lantai 3 ada 4 ruangan. Pada lantai 1 terdapat Ruangan Monitoring dengan jumlah komputer sebanyak 5 unit dan 1 *switch*, Ruangan Aplikasi 1 dengan jumlah komputer sebanyak 5 unit dan 1 *switch*, dan Ruangan Aplikasi 2 dengan jumlah komputer sebanyak 2 unit dan 1 *switch*. Sedangkan lantai 3 terdapat Ruangan Laboratorium A dengan jumlah komputer sebanyak 22 unit dan 1 *switch*, Ruangan Laboratorium B dengan jumlah komputer sebanyak 23 unit dan 1 *switch*, Laboratorium C dengan jumlah komputer sebanyak 22 unit dan 1 *switch*, Laboratorium D dengan jumlah komputer sebanyak 21 unit dan 1 *switch*. Pada lantai 3 ada Ruangan Server, terdapat perangkat jaringan dengan jumlah *switch* sebanyak 5 unit. Pada Ruangan Server tersebut ada 1 *switch* merupakan *switch* utama yang mem*forward* jaringan ke seluruh ruangan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau dengan ketentuan *bandhwidth* 1000 Mbps. Sedangkan komputer server terdapat di Ruangan Monitoring dengan sistem operasi *windows* 10."

Menurut Bapak Indra Mulia Syafutra, S.T. sebagai staff yang mengelola jaringan komputer di Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau mengatakan bahwa Pusat Teknologi dan Informasi Pangkalan Data pernah terjadinya serangan *Distributed Denial of Service* (DDoS). Sistem keamanan jaringan untuk melindungi terhadap serangan tersebut berupa *software* dan *hardware*. Sistem keamanan jaringan yang digunakan adalah *wazuh* sebagai *software* analasis jaringan. Terdapat kekurangan pada *wazuh* yaitu tidak bisa menyimpan *log* jaringan. Sedangkan *hardware firewall* yang digunakan adalah *fortigate*. Sistem keamanan menggunakan *fortigate* juga terdapat

kekurangan yaitu masa aktif penggunaan perangkat *firewall* yang terbatas dengan kode lisensi yang mahal.

Berdasarkan dari hasil wawancara yang telah dilakukan oleh peneliti, dibutuhkan sebuah perencanaan pembaruan untuk pengelolaan jaringan komputer yang lebih stabil dan optimal untuk Pusat Teknologi Informasi dan Pangkalan Data yaitu menggunakan metode *Software Defined Network* (SDN) dan dibutuhkan sistem keamanan berbasis *opensource* untuk keamanan jaringan di Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau yang mampu menyimpan *log* jaringan dan mengidentifikasi terhadap serangan *Distributed Denial of Service* (DDoS). Salah satu sistem keamanan jaringan yang mampu mendeteksi dan mencegah serangan DDoS yaitu *Intrusion Detection and Prevention System* menggunakan *snort.*

Penelitian terkait tentang "Perancangan *Intrusion Prevention System* pada Jaringan *Software Defined Networks* (SDN)". Penelitian melakukan simulasi pencegahan serangan *Denial of Service* (DoS) menggunakan *Intrusion Prevention System* (IPS) pada jaringan SDN. Penelitian ini melakukan simulasi pengujian serangan *Denial of Service* (DoS) menggunakan aplikasi *web Damn Vulnerable Web Application* (DVWA) pada kontroler *ryu*. Berdasarkan hasil penelitian menunjukkan bahwa IPS mampu meningkatkan keamanan jaringan SDN dengan ditandai pengaruh IPS yaitu terjadi penurunan kinerja terhadap *Quality of Service* dengan parameter *throughtput, delay, jitter*. Setiap kenaikkan 50 *rule* nilai *throughput* menurun sebesar 100 kbps, nilai *delay* naik sebesar 0,1 ms, dan nilai jitter naik sebesar 0,02 ms [4]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

Peneltian terkait tentang "Simulasi Pencegahan Serangan *Denial of Service* (DoS) Pada *Software Defined Network* Menggunakan *Intrusion Prevention System* (IPS) Dan Algoritma genetika". Peneltian mensimulasikan sistem pencegahan serangan *Denial of Service* (DoS) pada jaringan SDN menggunakan kontroler *ryu*. Berdasarkan hasil penelitian sistem IPS dan Algoritma genetika mampu memblokir serangan secara *realtime* dengan waktu *execute* sebesar 0,0278s[5]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

Penelitian tentang "*Integrasi Intrusion Prevention System* dan Analisa Performansi pada *Software Defined Network*". Peneltian ini melakukan dua pengujian serangan menggunakan DoS *Syn flood* dan DoS dengan paket ICMP. Dari hasil penelitiannya menunjukkan performa jaringan setelah dilakukan integrasi *Intrusion Prevention System*

(IPS) didalam jaringan SDN sangat stabil karena *Intrusion Prevention System* (IPS) mampu memblokir serangan DoS dan mampu menganalisa serangan yang masuk, ketika serangan memasuki 9000 *packet*/s mengalami penurunan[6]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

Berdasarkan kebutuhan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau dan penelitian terkait tentang sistem keamanan jaringan berbasis *opensource*, maka penulis berinisiatif untuk meneliti tentang **"Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection and Prevention System* (Studi Kasus: Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau)"**. Penelitian ini berfokus pada sistem keamanan *Intrusion Detection and Prevention System* (IDPS) menggunakan *snort* pada jaringan SDN dengan menyesuaikan topologi Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau. Simulasi rancangan menggunakan jaringan PTIPD UIN Suska Riau dan menggunakan sistem operasi linux Ubuntu yang sudah disetujui oleh Indra Mulia Syafutra, S.T sebagai staff yang mengelola jaringan komputer di PTIPD UIN Suska Riau.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka rumusan masalah dari penelitian ini adalah:

1. Bagaimana kinerja jaringan berbasis SDN dengan jaringan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau?
2. Bagaimana kinerja sistem keamanan *Intrusion Detection and Prevention System* (IDPS) menggunakan *snort* dalam mendeteksi dan mencegah serangan *Distributed Denial of Service* (DDoS) pada jaringan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau berbasis SDN?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk menganalisis kinerja sistem keamanan *Intrusion Detection and Prevention System* (IDPS) menggunakan *snort* pada jaringan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau berbasis *Software Defined Networks* (SDN) .

1.4 Batasan Masalah

Pada penelitian ini, peneliti membuat batasan masalah bertujuan untuk mencegah meluasnya ruang lingkup permasalahan dalam penelitian ini. Adapun batasan masalah tersebut diantaranya, yaitu:

1. Sistem operasi menggunakan Linux Ubuntu
2. Simulasi *Software Defined Network* tidak mendukung sistem operasi windows.
3. Arsitektur jaringan SDN dan sistem keamanan IDPS (*Intrusion Detection and Prevention System)* hanya berupa simulasi.
4. Simulasi rancangan jaringan berbasis SDN Menggunakan laptop.
5. Pengujian jaringan konvensional dan berbasis SDN hanya menggunakan *iperf3.*
6. Protokol yang diujikan hanya pada *Transmission Control Protocol* (TCP)
7. Sistem Keamanan jaringan IDPS menggunakan *snort,* dan *firewall* menggunakan *IPTables.*
8. Menangani serangan *Distributed Denial of Service* (DDoS) yaitu *Botnet attack* menggunakan *ufonet.*
9. Pengujian serangan DDoS dilakukan sebanyak 3 kali.
10. Pengujian *Quality of Service* (QoS) menggunkan *wireshark* dilakukan sebanyak 3 kali, parameter QoS yang diukur pada penelitian yaitu *throughput, delay, packet loss* .
11. Aplikasi *web* analisis serangan jaringan menggunakan *Basic Analysis and Security Engine* (BASE).

1.5 Manfaat Penelitian

Manfaat yang didapatkan pada penelitian ini adalah sebagai berikut:

1. Dengan adanya layanan jaringan berbasis *Software Defined Network* (SDN) tersebut memudahkan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau dalam mengelola jaringannya secara terpusat.
2. Sistem *Intrusion Detection and Prevention System* menggunakan *snort* mampu melindungi jaringan PTIPD UIN Suska Riau berbasis SDN terhadap serangan dan terjamin keamanannya serta dapat menyimpan *log* jaringan.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian TA (Tugas Akhir) ini yang berjudul "Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection and Prevention System* (IDPS) Dan *IPTables*" dengan Studi Kasus: Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau dilakukan studi literatur dengan melakukan pencarian teori serta referensi yang bersumber dari jurnal, paper, buku, internet dan sumber lainnya yang terkait dengan penelitian ini. Berikut ini beberapa penelitian terkait yang merupakan penelitian terdahulu dari referensi-referensi serta teori dari berbagai sumber dengan tujuan untuk menyelesaikan permasalahan pada penelitian ini.

Penelitian terkait membahas tentang "Analisis Dampak Serangan DDoS Pada Jaringan *Openflow*." Hasil penelitiannya menunjukkan bahwa penggunaan CPU dan memory meningkat secara signifikan dalam waktu 30 detik. Dampak serangan DDoS dapat mengganggu arus data dari user legal yang ada dalam jaringan untuk mengakses data tertentu karena layer kontrol dan pengiriman data sibuk melakukan processing data *dummy* (palsu) yang dikirimkan oleh *attacker*[2].

Penelitian terkait membahas tentang Dampak serangan DDoS pada *software based openflow switch* di perangkat HG553. Dalam penelitiannya menunjukkan UDP *Flooding* yang terjadi dalam jaringan *Software Defined Network* menyebabkan jaringan *Software Defined Network* sulit untuk di akses dikarenakan *Switch Opeflow* terlepas dari kontroller dan menyebabkan penggunaan *reource* tinggi pada *Switch Openflow*[3].

Penelitian terkait tentang "Perancangan *Intrusion Prevention System* pada Jaringan *Software Defined Networks* (SDN)". Penelitian melakukan simulasi pencegahan serangan *Denial of Service* (DoS) menggunakan *Intrusion Prevention System* (IPS) pada jaringan SDN. Penelitian ini melakukan simulasi pengujian serangan *Denial of Service* (DoS) menggunakan aplikasi *web Damn Vulnerable Web Application* (DVWA) pada kontroler *ryu*. Berdasarkan hasil penelitian menunjukkan bahwa *Intrusion Prevention System* (IPS) mampu meningkatkan keamanan jaringan SDN dengan ditandai pengaruh sistem IPS yaitu terjadi penurunan kinerja terhadap *Quality of Service* dengan parameter *throughtput, delay,*

Jitter. Setiap kenaikkan 50 *rule* nilai *throughput* menurun sebesar 100 kbps, nilai *delay* naik sebesar 0,1 ms, dan nilai jitter naik sebesar 0,02 ms [4]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

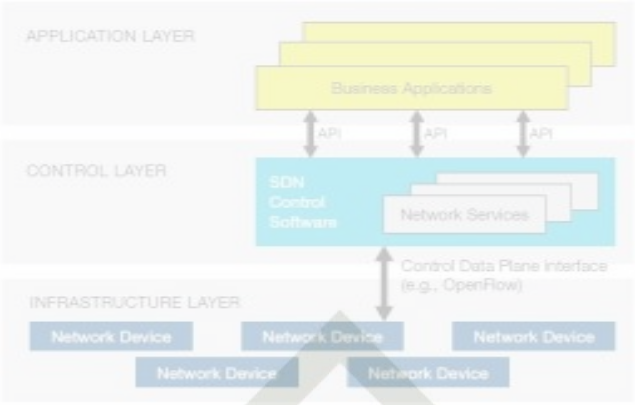
Peneltian terkait tentang "Simulasi Pencegahan Serangan *Denial of Service* (DoS) pada *Software Defined Network* (SDN) Menggunakan *Intrusion Prevention System* (IPS) Dan Algoritma genetika". Peneltian mensimulasikan sistem pencegahan serangan *Denial of Service* (DoS) pada jaringan SDN menggunakan kontroler *ryu*. Berdasarkan hasil penelitian *sistem* IPS dan Algoritma genetika mampu memblokir serangan secara *realtime* dengan waktu *execute* sebesar 0,0278s[5]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

Penelitian tentang "*Integrasi Intrusion Prevention System* dan Analisa Performansi pada *Software Defined Network*(SDN)". Peneltian ini melakukan dua pengujian serangan menggunakan DoS *Syn flood* dan DoS dengan paket ICMP. Dari hasil penelitiannya menunjukkan performa jaringan setelah dilakukan integrasi IPS didalam jaringan SDN sangat stabil karena IPS mampu memblokir serangan DoS dan mampu menganalisa serangan yang masuk, ketika serangan memasuki 9000 *packet*/s mengalami penurunan[6]. Penelitian ini tidak dilengkapi dengan sistem IDS (*Intrusion Detection System*).

2.2 *Software Defined Network* (SDN)

Software Defined Network (SDN) adalah sebuah paradigma arsitektur baru dalam bidang jaringan komputer untuk mendesain, membangun dan mengelola jaringan yang bersifat d*ynamic, manageable, cost-effective, dan adaptable.* Konsep utama SDN yaitu memisahkan secara fisik *control plane* dengan data *plane*, sehingga kontroler dapat diprogram secara langsung (*directly programmable*), sedangkan infrastruktur yang mendasarinya dapat diabstraksikan untuk *layer application* dan *network services*[7].

Ada 3 bidang utama pada arsitektur SDN diantaranya *application layer*, *control layer*, dan *infrastructure layer*. Seperti ditunjukkan pada Gambar 2.1 dibawah ini.



Gambar 2.1 Arsitektur SDN

1. Lapisan Infrastruktur (*infrastructure layer*) atau data *plane*

   Lapisan ini berisi elemen-elemen jaringan yang dapat mengatur seluruh data path dari SDN sesuai dengan instruksi dari *Control Data Plane Interface* (CDPI).

2. Lapisan Kontrol (*control layer*) atau *control plane*

   Lapisan ini dapat mentranslasikan kebutuhan dari aplikasi disertai infrastruktur dengan memberi instruksi yang sesuai untuk data path dari SDN. Selain itu, lapisan ini dapat memberikan informasi yang tepat dan dibutuhkan oleh aplikasi SDN.

3. Lapisan Aplikasi (*application layer)* atau application plane

   Lapisan ini terdapat pada lapisan terluar atau teratas pada SDN, fungsi dari lapisan ini yaitu dapat berkomunikasi dengan sistem melalui *NorthBound Interface* (NBI).

2.3 *OpenFlow*

OpenFlow adalah protokol paling utama pada *Software Defined Network* (SDN) posisinya berada di antara *controller* dan *forwarding (data plane)*. *OpenFlow* memungkinkan pengaturan *routing* dan pengiriman paket ketika melalui sebuah *switch*. Dalam sebuah jaringan, setiap *switch* hanya berfungsi meneruskan paket yang melalui suatu *port* tanpa mampu membedakan tipe protokol data yang dikirimkan *OpenFlow* memungkinkan untuk mengakses dan memanipulasi *forwarding plane* secara langsung dari perangkat-perangkat jaringan seperti *switch* dan *router* baik secara fisik maupun *virtual*[7].

2.4 *Mininet*

Mininet merupakan sebuah emulator pada Jaringan *Software Defined Network* (SDN), yang dapat mensimulasikan kinerja sebuah jaringan yang telah dirancang dalam sebuah kernel *Linux*. *Mininet* diciptakan dengan tujuan untuk mendukung riset di bidang *Software Defined Network* (SDN) dan *OpenFlow*. Emulator *Mininet* memungkinkan kita untuk menjalankan sebuah kode secara interaktif di atas laptop atau di atas virtual *hardware*, tanpa harus memodifikasi kode tersebut. Artinya kode simulasi sama persis dengan kode pada *real network environment*[7]

Untuk melakukan pengujian sambungan pada mininet dapat dilakukan dengan perintah (*command*) "sudo mn". Dengan *command* ini *mininet* akan mengemulasikan konfigurasi jaringan SDN yang terdiri dari 1 kontroler, 1 *switch* dan 2 *host*.

Gambar 2.2 *Single Command* pada *mininet*

.

2.5 Perbandingan Jaringan konvensional dengan Jaringan berbasis *Software Defined Network* (SDN)

Adapun perbandingan antara jaringan konvensional dengan jaringan *Software Defined Network* (SDN) adalah:

1. Konsep utama jaringan konvensional adalah kontroler dan *forwarding plane* terletak dalam satu perangkat *network* yang sama dan tidak terpisah, namun saling terhubung dengan perangkat jaringan komputer yang lainnya. Untuk konfigurasi jaringan konvensional harus secara manual dengan mengkonfigurasi masing-masing perangkat jaringan seperti Gambar 2.3 berikut ini:

Gambar 2.3 Topologi Konvensional[8]

2. Konsep utama jaringan *Software Defined Network* (SDN) dengan adanya protokol *openflow* adalah memisahkan secara fisik anatara sistem kontrol dan *forwarding plane*. Dimana *forwarding plane* terdapat pada perangkat jaringan yaitu *switch* dan *router*, sedangkan kontroler terpisah dari perangkat jaringan seperti Gambar 2.4 berikut ini[9]:

Gambar 2.4 Topologi *Software Defined Network* (SDN) [8]

2.6 *Open Network Operating System* (ONOS)

Open Network Operating System (ONOS) merupakan Sistem Operasi jaringan *Software Defined Network* (SDN) bersifat *open source* yang pertama berbasis Java, tujuan utama ONOS adalah sebagai *service provider network*. ONOS telah dilengkapi tampilan GUI (*Graphical User Interface*) untuk mempermudah kontrol terhadap perangkat yang mendukung protokol *Openflow* yang terhubung padanya. Setiap fitur dalam ONOS diaktifkan melalui *Apache Karaf* (OSGi *runtime*). ONOS dirancang untuk memenuhi kebutuhan operator dengan menerapkan layanan jaringan yang dinamis dengan antarmuka

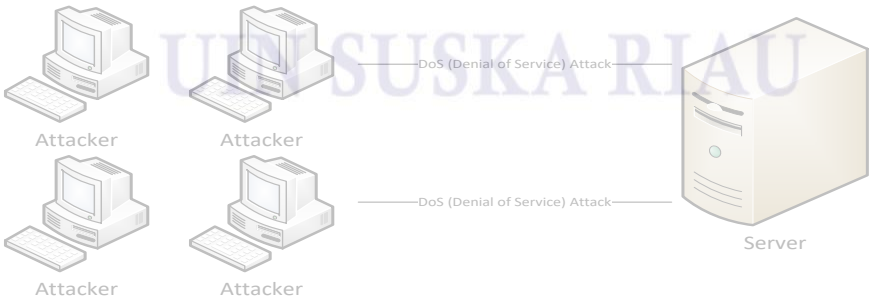
rogram yang disederhanakan. ONOS mendukung konfigurasi dan kontrol jaringan secara *real-time*, mudah membuat jaringan baru tanpa perlu mengubah sistem data *plan*[10].

2.7 Perbedaan DoS dan DDoS

Denial of Service (DoS) merupakan serangan yang melibatkan satu komputer / koneksi internet untuk *flood* (membanjiri) sebuah server dengan paket (tcp/UDP) - Tujuan dari serangan ini adalah untuk "*overload*" *bandwith* server atau sumber lainnya. Ini menyebabkan *Denial Of Servis* kepada orang lain yang berusaha untuk menggunakan server tersebut untuk alasan tertentu[11].

Gambar 2.5 *Denial of Service* (DoS) *Attack*

Distributed Denial of Service (DDoS) bertujuan untuk menghambat kinerja komputer dalam segi sumber daya dalam jaringan *user* dengan membanjiri sumber daya berupa paket data yang tidak berguna sehingga melemahkan sumber daya didalam jaringan dengan sejumlah besar *traffic*. Secara umum serangan DDoS terdiri dari beberapa jenis, serangan dengan basis *bandwidth*, serangan dengan basis lalu lintas jaringan, dan serangan dengan basis aplikasi.



Gambar 2.6 *Distributed Denial of Service* (DDoS)

Perbedaan DOS dan DDOS *attack* adalah:

1. Bahwa penyerang hanya menggunakan satu komputer dan satu sambungan Internet saat meluncurkan serangan DoS.
2. Penyerang menggunakan jaringan komputer yang tersebar luas dan banyak sambungan internet dalam serangan DDoS.
3. Serangan DoS jauh lebih mudah dioperasikan dan biaya yang lebih rendah.
4. Lebih sulit untuk menahan serangan DDoS karena ada banyak sumber yang mengirimkan permintaan untuk membanjiri sistem target. Dalam hal ini, memblokir sumber hampir tidak mungkin dilakukan.
5. Dalam serangan DoS, jika lalu lintas masuk dikenali sebagai lonjakan lalu lintas yang tidak normal, *host* dapat mengambil tindakan segera untuk memblokir sumber. Artinya, serangan DoS dapat diblokir dalam waktu yang relatif singkat.

2.8 *BotNet (Robot Network)* dan *Ufonet*

BotNet adalah sebuah *zombie* dalam jaringan dari jutaan perangkat yang tersambung ke internet, yang mana *bot* diinfeksi dengan *malware* yang khusus agar bias dikendalikan oleh *cybercriminal* dari jarak jauh untuk memberikan serangan seperti mengirim email, mencuri informasi pribadi, dan meluncurkan serangan DDoS[12]. UFONet merupakan sebuah *software* gratis, P2P (Peer to Peer) dan kriptografi,. UFONet juga perangkat lunak yang memungkinkan untuk melakukan serangan DoS dan DDoS ke target melalui eksploitasi vektor *Open Redirect*.

2.9 *Intrusion Detection System* (IDS)

Intrusion Detection System (IDS) adalah sebuah sistem yang melakukan pengawasan terhadap *traffic* jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan. Jika ditemukan kegiatan-kegiatan yang mencurigakan berhubungan dengan trafficjaringan maka *Intrusion Detection System* (IDS) akan memberikan peringatan kepada sistem atau administratorjaringan[13].

Ada *Intrusion Detection System* (IDS) yang fungsinya hanya sebagai pengawas dan pemberi peringatan ketika terjadi serangan dan ada juga *Intrusion Detection System* (IDS) yang bekerja tidak hanya sebagai pengawas dan pemberi peringatan melainkan juga dapat

melakukan sebuah kegiatan yang merespon adanya percobaan serangan terhadap sistem jaringan dan komputer. *Intrusion Detection System* (IDS) sendiri dibedakan menjadi 2 jenis, yaitu :

1. *Network Intrusion Detection System (NIDS)*

   Intrusion Detection System jenis ini ditempatkan disebuah tempat/ titik yang strategis atau sebuah titik didalam sebuah jaringan untuk melakukan pengawasan terhadap trafik yang menuju dan berasal dari semua alat-alat (*devices*) dalam jaringan. Idealnya semua trafik yang berasal dari luar dan dalam jaringan di lakukan di *scan*, namun cara ini dapat menyebabkan *bottleneck* yang mengganggu kecepatan akses di seluruh jaringan.

2. *Host Intrusion Detection System (HIDS)*

   Intrusion Detection System (IDS) jenis ini berjalan pada host yang berdiri sendiri atau perlengkapan dalam sebuah jaringan. Sebuah *Host Intrusion Detection System* (HIDS) melakukan pengawasan terhadap paket-paket yang berasal dari dalam maupun dari luar hanya pada satu alat saja dan kemudian memberi peringatan kepada *user* atau administratorsistem jaringan akan adanya kegiatan-kegiatan yang mencurigakan yang terdeteksi oleh *Host Intrusion Detection System* (HIDS).

2.10 *Intrusion Prevention System* (IPS)

Intrusion Prevention System (IPS) merupakan suatu cara ataupun metode untuk mencegah serangan yang akan masuk ke jaringan komputer pengguna, dengan memeriksa semua paket yang masuk dan apabila teridentifikasi serangan, IPS akan mem*block* dan mencatat *log* semua paket yang teridentifikasi. Pendekatan yang sering digunakan system keamanan komputer, IPS mengkombinasikan teknik *firewall* dan metode *Intrusion Detection System* (IDS) dengan sangat baik. Jadi *Intrusion Prevention System* (IPS) disini bertindak seperti layaknya *firewall* yang akan melakukan *allow* dan *block* yang dikombinasikan seperti IDS yang dapat mendeteksi paket secara detail. IPS menggunakan *signatures* untuk mendeteksi di aktifitas trafik di jaringan dan terminal, dimana pendeteksian paket yang masuk dan keluar dapat di cegah sedini mungkin sebelum merusak atau mendapatkan akses ke dalam jaringan lokal[13].

Secara umum, ada dua jenis IPS, yaitu *Host-based Intrusion Prevention System* (HIPS) dan *Network-based Intrusion Prevention System* (NIPS).

1. *Host-based Intrusion Prevention System* (HIPS)

   Host-based Intrusion Prevention System (HIPS) adalah sama seperti halnya *Host-based Intrusion Detection System* (HIDS). Program agen HIPS diinstall secara langsung di sistem yang diproteksi untuk dimonitor aktifitas sistem internalnya. HIPS di-binding dengan kernel sistem operasi dan *services* sistem operasi. Sehingga HIPS bisa memantau dan menghadang system call yang dicurigai dalam rangka mencegah terjadinya intrusi terhadap *host*. HIPS juga bisa memantau aliran data dan aktivitas pada aplikasi tertentu. Sebagai contoh HIPS untuk mencegah *intrusion* pada *web server* misalnya. Dari sisi *security* mungkin solusi HIPS bisa mencegah datangnya ancaman terhadap host. Tetapi dari sisi *performance*, harus diperhatikan apakah HIPS memberikan dampak negatif terhadap *performance host*. Karena menginstall dan binding HIPS pada sistem operasi mengakibatkan penggunaan *resource* komputer host menjadi semakin besar.

2. *Network-based Intrusion Prevention System* (NIPS)

   Network-based Intrusion Prevention System (NIPS) tidak melakukan pantauan secara khusus di satu *host* saja. Tetapi melakukan pantauan dan proteksi dalam satu jaringan secara *global*. NIPS menggabungkan fitur IPS dengan *firewall* dan kadang disebut sebagai *In-Line* IDS atau *Gateway Intrusion Detection System* (GIDS). Sistem kerja IPS yang populer yaitu pendeteksian berbasis *signature*, pendeteksian berbasis anomali, dan monitoring berkas-berkas pada sistem operasi *host*.

2.11 *Snort*

Snort merupakan sebuah aplikasi *tool security* yang berfungsi untuk mendeteksi intrusi jaringan yang kemungkinan terjadinya serangan dan melakukan pencegahan. *Snort* dikonfigurasikan untuk mengawasi jaringan dari jenis serangan[14].

Ada 3 fungsi utama pada *snort* yaitu:

1. Penangkal program-program *sniffer* paket-paket (seperti tcpdump).
2. *Packet logger* (berguna untuk men-*debug* trafik-trafik jaringan).
3. Sistem pencegah intrusi untuk sistem jaringan.

snort dapat dikonfigurasikan untuk berjalan pada *mode-mode* berikut ini :

1. *Sniffer mode*

   Berfungsi membaca paket dalam jaringan dan menampilkan dalam bentuk aliran tidak terputus pada konsol (layar).

2. *Packet logger mode*

   Berfungsi mencatat *log* dari paket ke dalam *disk*.

3. *Network Intrussion Detection System* (NIDS) *mode*

   Memiliki konfigurasi kompleks, namun bisa dimodifikasi, yang membuat *snort* bisa menganalisa arus jaringan untuk membandingkan dengan rangkaian *ruleset* yang dibuat *user*, sekaligus melakukan beberapa tindakan berdasarkan hal yang diamatinya.

4. *Inline mode*

   Berfungsi mengambil paket dari *iptables* (daripada *libpcap*) dan menginstruksikan *iptables* untuk menolak atau meneruskan paket tersebut berdasarkan jenis *rule* dari *snort* yang digunakan.

Snort memenuhi kriteria tersebut, yaitu dapat dikonfigurasi dan dibiarkan berjalan untuk periode yang lama tanpa meminta pengawasan atau perawatan bersifat administratif sebagai bagian dari sistem keamanan terpadu sebuah infrastruktur jaringan.

2.12 *Basic Analysis Security Engine* (BASE)

Basic Analysis Security Engine (BASE) adalah sebuah *interface web* untuk melakukan analisis dari intrusi yang *snort* telah deteksi pada jaringan. *Basic Analysis Security Engine* (BASE) ditulis oleh kevin johnson adalah program analisis sistem jaringan berbasis PHP yang mencari dan memproses database dari *security event* yang dihasilkan oleh berbagai program monitoring jaringan, *firewall*, atau sensor IDS. Berikut ini adalah beberapa kelebihan dari *Basic Analysis Security Engine* (BASE) yaitu :

Program berbasis *web* yang memungkinkan implementasi antar platform.

Log-log yang sulit untuk dibaca akan menjadi mudah untuk dibaca.

Data-data dapat dicari sesuai dengan kriteria tertentu.

Opensource yang merupakan perintis antarmuka GUI untuk snort dan paling banyak digunakan oleh pengguna *Intrusion Detection System* (IDS). *Basic Analysis Security Engine* (BASE) merupakan rekomendasi dari Snort.org sendiri.

5. Multi *language*, antarmuka memilki beberapa bahasa selain bahasa Inggris dan layanan peringatan yang real time.
6. Dapat diimplementasikan pada *Intrusion Detection System* (IDS) manapun selain snort.

Berikut ini adalah beberapa Fitur yang ada pada *Basic Analysis Security Engine* (BASE)

1. Ditulis dalam bahasa PHP.
2. Menganalisa *log* intrusi.
3. Men*display* informasi *database* dalam bentuk *web*.
4. Mengenerate *graph* dan *alert* berdasarkan sensor, waktu *rule* dan *protocol*.
5. Men*display* summary *log* dari semua alertdan linkuntuk *graph*.
6. Dapat diatur berdasarkan kategori grup *alert*, *false* positif dan *e-mail*.

2.13 *Iptables*

Iptables adalah suatu *tools* dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan filter terhadap *traffic* lalulintas data. Secara sederhana digambarkan sebagai pengatur lalu lintas data. Dengan *iptables* guna untuk mengatur semua lalu lintas dalam komputer, baik yang masuk ke komputer, keluar dari komputer, ataupun *traffic* yang sekedar melewati komputer.

2.14 *Wireshark*

Wireshark adalah sebuah *tools* jaringan yang ditujukan untuk menganalisa paket data jaringan. *Wireshark* disebut juga *Network Packet Analyzer* yang memiliki fungsi untuk menangkap paket-paket jaringan dan untuk menampilkan semua informasi paket yang terdeteksi sedetail mungkin. *Network Packet Analyzer* sebagai alat untuk memeriksa kejadian yang terjadi di dalam jaringan baik pada kabel maupun *wireless*. Dengan adanya *wireshark* ini semua sangat dimudahkan dalam hal memonitoring dan menganalisa paket yang lewat di jaringan.

Ada beberapa contoh penggunaan *Wireshark* [15].

1. Admin sebuah jaringan menggunakan untuk *troubleshooting* masalah di jaringan.

2. Admin menggunakan *Wireshark* sebagai sistem keamanan jaringan. Beberapa fitur kelebihan *Wireshark*, diantaranya :
   1. Berjalan pada *Operating System* (OS) *Linux* dan *Windows*.
   2. Menangkap paket (*Capturing Packet*) langsung dari jaringan antarmuka (*network interface*).
   3. Mampu menampilkan hasil tangkapan dengan detail.
   4. Dapat melakukan pem*filter*an paket.
   5. Hasil tangkapan dapat di *save*, di *import* dan di *export*.

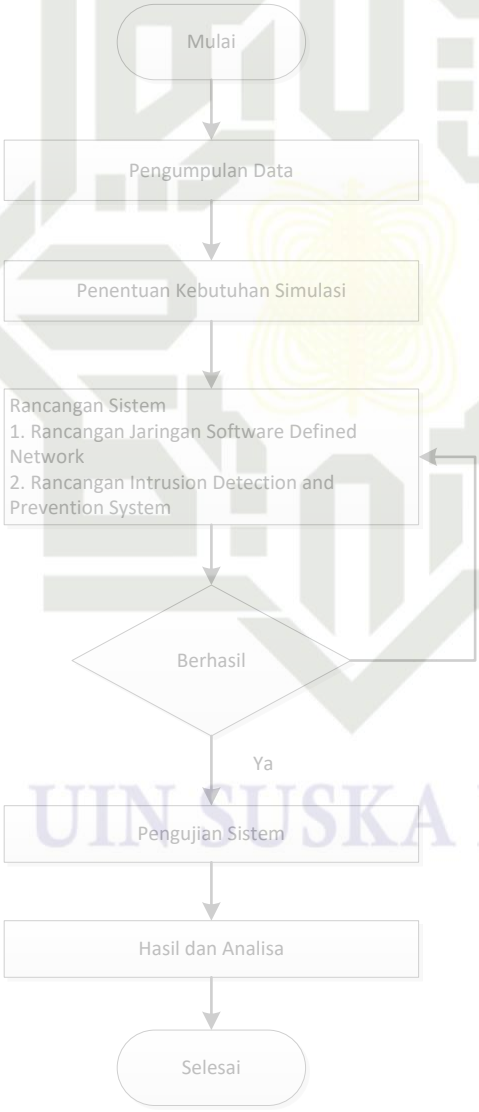
2.15 *Quality of Service* (*QoS*)

Quality of Service (*QoS*) merupakan mekanisme pada jaringan yang menentukan bahwa suatu *service* berupa aplikasi-aplikasi dapat beroperasi sesuai dengan standart kualitas *service* yang telah diterapkan. Parameter-parameter *Quality of Service* (*QoS*) seperti *troughput*, *Delay*, dan *packetloss*.

BAB III
METODOLOGI PENELITIAN

3.1 Alur Tahapan Penelitian

Pada bab metodologi penelitian ini akan dibahas tahapan dan langkah-langkah penulis dalam melakukan penelitian mulai dari awal penelitian sampai akhir penelitian. Adapun tahapan-tahapan penelitian yang akan dilakukan seperti ditunjukkan Gambar 3.1 berikut.



Gambar 3.1 Alur Tahapan Penelitian

3.2 Pengumpulan Data

Pengumpulan data guna untuk memperoleh informasi dalam mendukung prosedur penelitian yang belum diketahui sehingga penelitian menjadi layak untuk dilakukan. Pengumpulan data menjadi bahan acuan dan langkah awal dalam penulisan Tugas Akhir (TA) yang berjudul "Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection and Prevention System* (IDPS) (Studi Kasus: Pusat Teknologi Informasi Pangkalan Data UIN Suska Riau). Pengumpulan data dilakukan dengan melakukan wawancara bersama Bapak Indra Mulia Syafutra, S.T yaitu *staff* Pusat Teknologi Informasi dan Pangkalan Data sebagai pengelola jaringan di UIN Suska Riau, dengan tujuan untuk mencari tahu informasi tentang jaringan yang ada di Pusat Teknologi Informasi Pangkalan Data UIN Suska Riau seperti topologi jaringan, operating system, perangkat jaringan dan sistem keamanan jaringan yang digunakan.

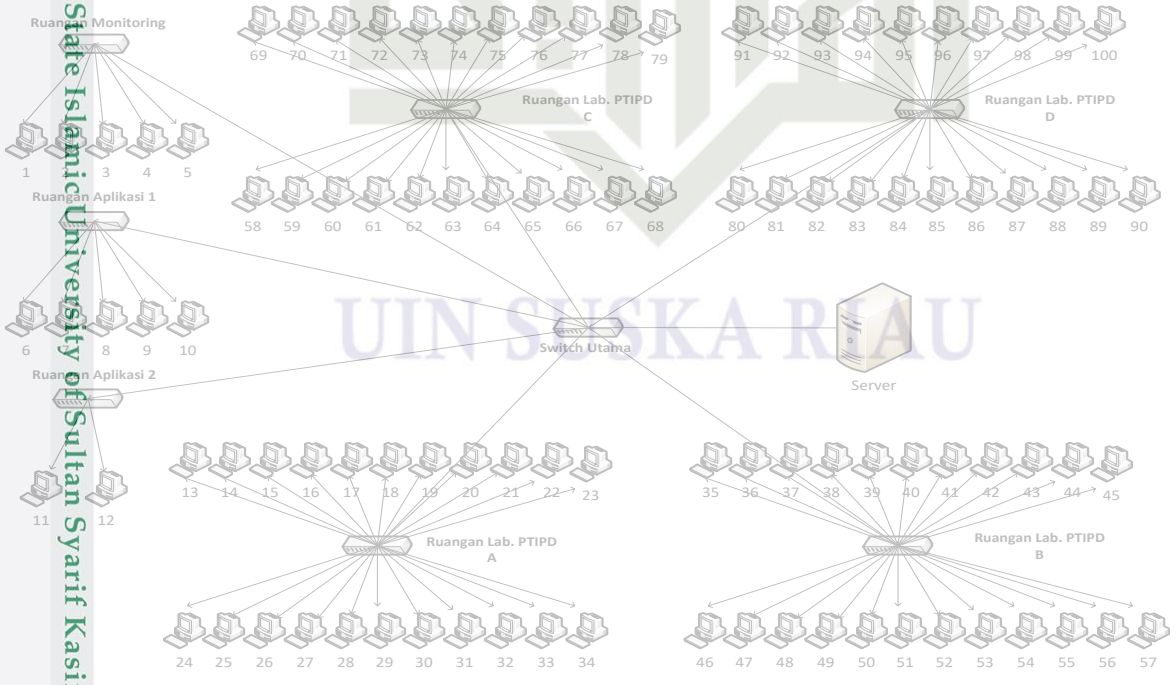
Berdasarkan hasil wawancara bersama *staff* Pusat Teknologi Informasi dan Pangkalan Data serta pengelola jaringan di UIN Suska Riau yaitu Bapak Indra Mulia Syafutra, S.T. beliau mengatakan bahwa "Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau menggunakan topologi star terdiri dari 3 lantai dan terdapat ruangan yang menggunakan jaringan komputer dengan jumlah keseluruhan 7 ruangan yaitu pada lantai 1 ada 3 ruangan dan lantai 3 ada 4 ruangan. Pada lantai 1 terdapat Ruangan Monitoring dengan jumlah komputer sebanyak 5 unit dan 1 *switch*, Ruangan Aplikasi 1 dengan jumlah komputer sebanyak 5 unit dan 1 *switch*, dan Ruangan Aplikasi 2 dengan jumlah komputer sebanyak 2 unit dan 1 *switch*. Sedangkan lantai 3 terdapat Ruangan Laboratorium A dengan jumlah komputer sebanyak 22 unit dan 1 *switch*, Ruangan Laboratorium B dengan jumlah komputer sebanyak 23 unit dan 1 *switch*, Laboratorium C dengan jumlah komputer sebanyak 22 unit dan 1 *switch*, Laboratorium D dengan jumlah komputer sebanyak 21 unit dan 1 *switch*". Pada lantai 3 ada Ruangan Server, terdapat perangkat jaringan dengan jumlah *switch* sebanyak 5 unit. Pada Ruangan Server ada 1 *switch* merupakan *switch* utama yang mem*forward* jaringan ke seluruh ruangan Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau.

Menurut Bapak Indra Mulia Syafutra, S.T. sebagai staff yang mengelola jaringan komputer di Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau mengatakan bahwa Pusat Teknologi dan Informasi Pangkalan Data menggunakan sistem keamanan jaringan berupa *software* dan *hardware*. Sistem keamanan jaringan yang digunakan adalah *wazuh* sebagai *software* analasis jaringan. Terdapat kekurangan pada *wazuh* yaitu tidak

bisa menyimpan *log* jaringan. Sedangkan *hardware firewall* yang digunakan adalah *fortigate*. Sistem keamanan menggunakan *fortigate* juga terdapat kekurangan yaitu masa aktif perangkat *firewall* yang terbatas. Sehingga *administartor network* Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau harus membeli kode lisensi yang mahal untuk memperpanjang masa pemakaian sistem keamanan *fortigate*. Menurut Bapak Indra Mulia Syafutra, S.T dibutuhkan sisitem keamanan jaringan yang berbasis *opensource* yang namanya melindungi jaringan di Pusat Teknologi dan Informasi Pangkalan Data UIN Suska Riau.

3.2.1 Topologi Jaringan Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau

Berikut ini adalah gambaran umum *skema* topologi jaringan yang diterapkan pada Pusat Teknologi Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada saat ini. Informasi ini didapatkan pada saat melakukan wawancara dengan Bapak Indra Mulia Syafutra, S.T yaitu *staff* Pusat Teknologi Informasi dan Pangkalan Data sebagai pengelola jaringan di UIN Suska Riau, yaitu menggunakan topologi jaringan *star*, seperti terlihat pada Gambar 3.2 dibawah ini.



Gambar 3.2 Topologi Jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau

Berikut ini spesifikasi *Hardware* dan *software* yang digunakan server di PTIPD UIN Suska Riau.

Personal *Computer* (PC) Sangfor, Teknologi VM dari Sangfor HCI

CPU :

14 *core*(s) 28 *Thread* X 2 (*Intel® Xeon® Gold* 6132 CPU @ 2.60 GHz)

b) *Clock Speed* : 2.60 GHz
c) *Chace* : 19.25 MB
d) *Stepping* : 4
e) *Node* : Ada 5
f) Memory : 256 GB
g) *Harddisk* : SSD 128 GB
h) *Harddisk Network* : 300 TB
i) *Operating System* : Multi OS (*windows*, *Ubuntu*, *debian* dan *centos*)

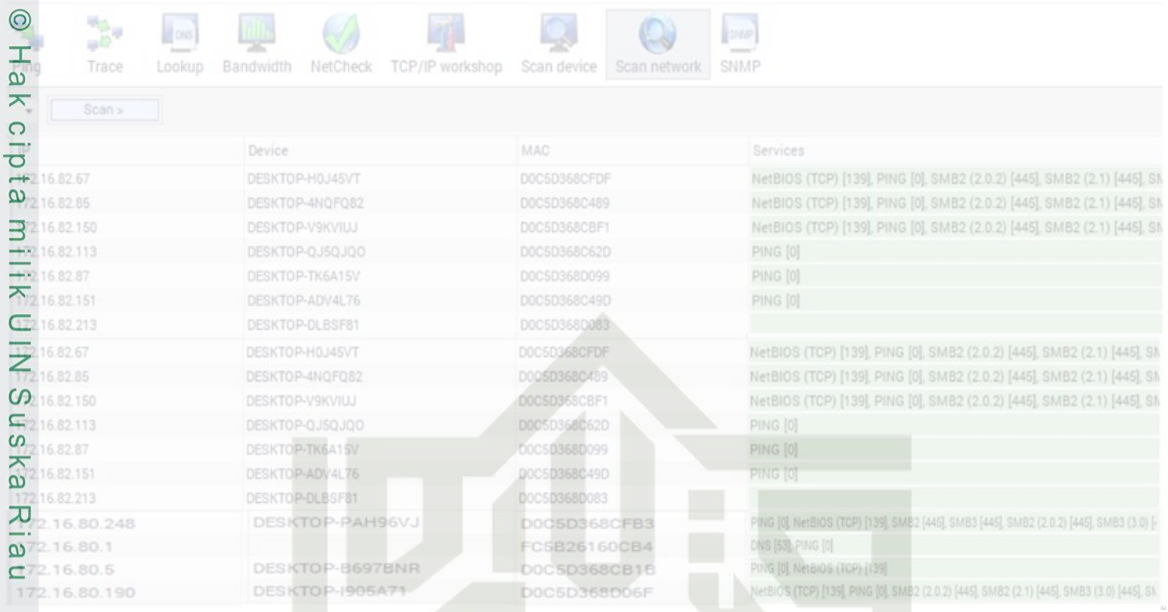
2. *Router* : MikroBit Dinara
   a) *Processor* : *Intel Xeon Hexa Core* 3.7GHz
   b) *Chipset* : *Intel* C246
   c) RAM

   x DDR4-266 U-DIMM *Dual-Channel*, *up to* 32 GB non-ECC/E memory, 8 GB DDR4 2400 ECC *Installed*
   d) *Network Port*

   8 *port intel* Gigabit *Ethernet*, 8 *port Intel* Gigabit SFP, 2 *port Intel* 10G SFP+
   e) PSU : 300 *Watt Redundant* PSU
   f) *Operating System* : *MikroTik router* OS
3. *Firewall* : *Fortigate*
4. *Analysis* Jaringan : *wazuh*

3.2.2 IP Address Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau

Berikut ini adalah beberapa ip address yang terdeteksi didalam jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau yang ditampilkan menggunakan aplikasi *Axence NetTolls*, antara lain sebagai berikut.



Gambar 3.3 Tampilan IP Address Pusat Teknologi Informasi Dan Pangkalan Data (PTIPD) UIN Suska Riau

3.3 Penentuan Kebutuhan Simulasi

Pada tahapan ini peneliti didasari dengan konsep utama *Software Defined Network* (SDN) yaitu memisahkan antara *control plane* dan data *plane*, maka peneliti menggunakan kontroler SDN yaitu *Open Network Operating System* (ONOS) dipisah dengan *Mininet* yang akan berperan dalam menjalankan data *plane* pada sistem operasi *Linux Ubuntu* karena *mininet* dirancang dalam sebuah kernel *Linux.* ONOS berfungsi sebagai pengontrol dan *Mininet* berfungsi sebagai sebuah *emulator* berbasis CLI yang digunakan untuk membuat sebuah topologi jaringan berbasis SDN yang menciptakan jaringan *virtual* yang *realistis*, berjalan dengan *kernel* yang sama, *switch* dan kode aplikasi pada mesin yang sama pada aslinya. Dalam Tugas Akhir ini simulasi menggunakan jaringan Pusat Teknologi Informasi Dan Pangkalan Data (PTIPD) UIN Suska Riau dilaksanakan di Ruangan Lab. PTIPD B. Pada penelitian ini dibutuhkan alat dan bahan untuk membantu penulis dalam simulasi jaringan SDN. Kebutuhan tersebut meliputi kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai berikut:

Spesifikasi perangkat keras (*Hardware*):

1. Laptop Acer Aspire 4738Z 14” resolusi 1366 x 768 *pixels*
2. *Processor* : Intel(R) Core(TM) i3 2.27 GHz
3. RAM (*Random Access Memory*) : 4,00 GB
4. *Harddisk* : 500.1 GB

Spesifikasi perangkat lunak (*software*):

1. *Operating System* (OS) : *Ubuntu* 18.04
2. Emulator SDN :*Mininet* versi 2.3.0
3. Kontroler SDN :*Open Network Operating System*
4. Aplikasi Pengukuran *Quality of Service* :*Wireshark*
5. *Intrusion Detection and Prevention System* : *Snort* versi 2.9.17.1
6. *Victim* : *Operating System* kalilinux 2020.3
7. *Distributed Denial of Service* : *Ufonet*
8. Aplikasi Pengukuran *Quality of Service* :*Wireshark*
9. *Analysis* Jaringan : *Basic Analysis and Security Engine*

3.4 Rancangan Sistem

3.4.1 Rancangan Jaringan *Software Defined Network* (SDN)

Tahap perancangan *Software Defined Network* (SDN) adalah bagaimana merancang komponen-komponen yang akan digunakan didalam jaringan SDN pada penelitian. Rancangan jaringan SDN dibuat berdasarkan topologi yang ada di Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau. Adapun rancangan arsitektur serta komponen-komponen yang digunakan dalam jaringan SDN adalah sebagai berikut.

1. *Software Defined Network* (SDN) *Controller*

   Software Defined Network (SDN) *Controller* yang digunakan adalah ONOS *controller* yang berfungsi untuk mengontrol seluruh lalu lintas Jaringan SDN.

2. *Intrusion Detection and Prevention System* (IDPS)

   Intrusion Detection and Prevention System (IDPS) menggunakan *snort* berfungsi untuk monitoring serta mendeteksi dan mencegah serangan.

3. *Open vSwitch*

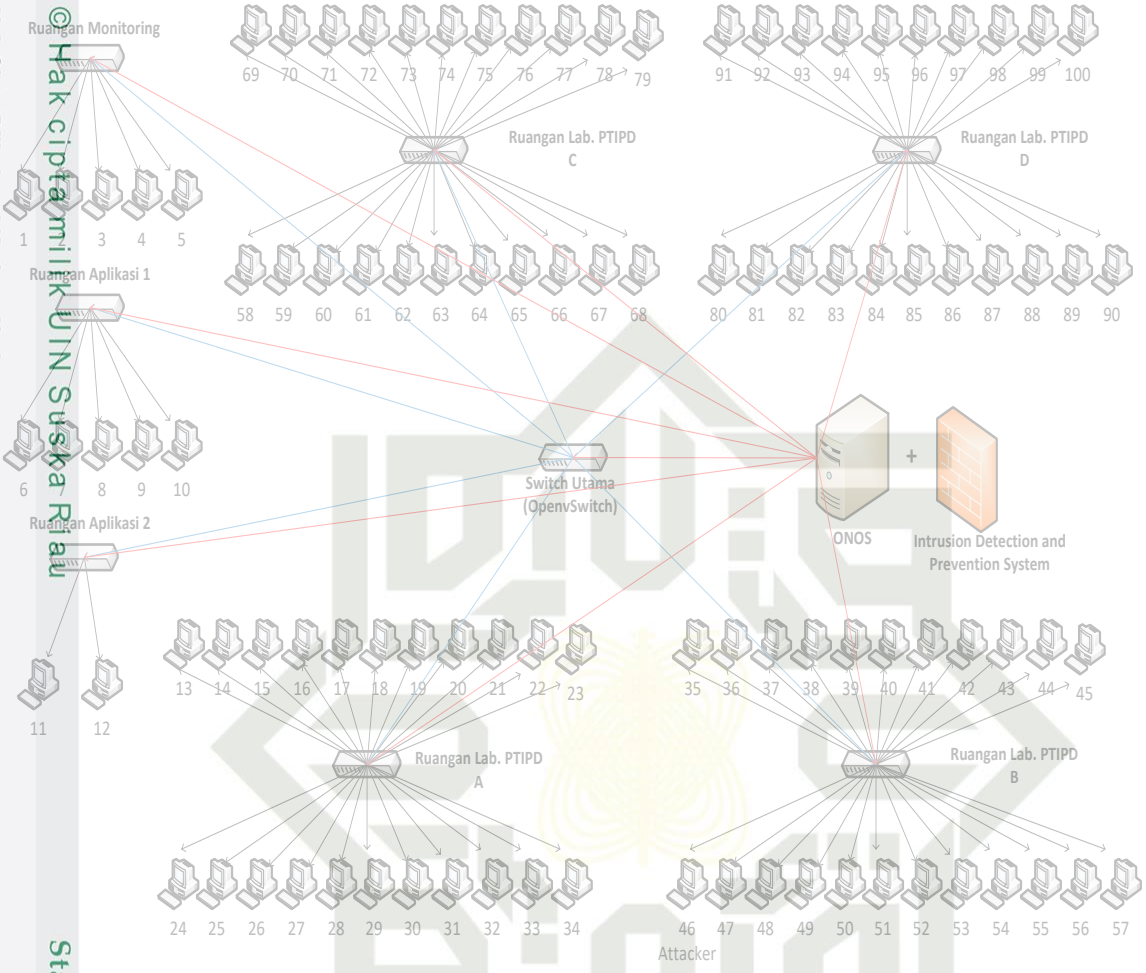
   Switch yang digunakan telah didukung *protocol openflow.*

4. *User*

   User sebuah komputer yang terhubung didalam jaringan SDN.

5. *Attacker*

   User yang melakukan serangan DDoS yaitu *botnet attack* pada jaringan SDN.



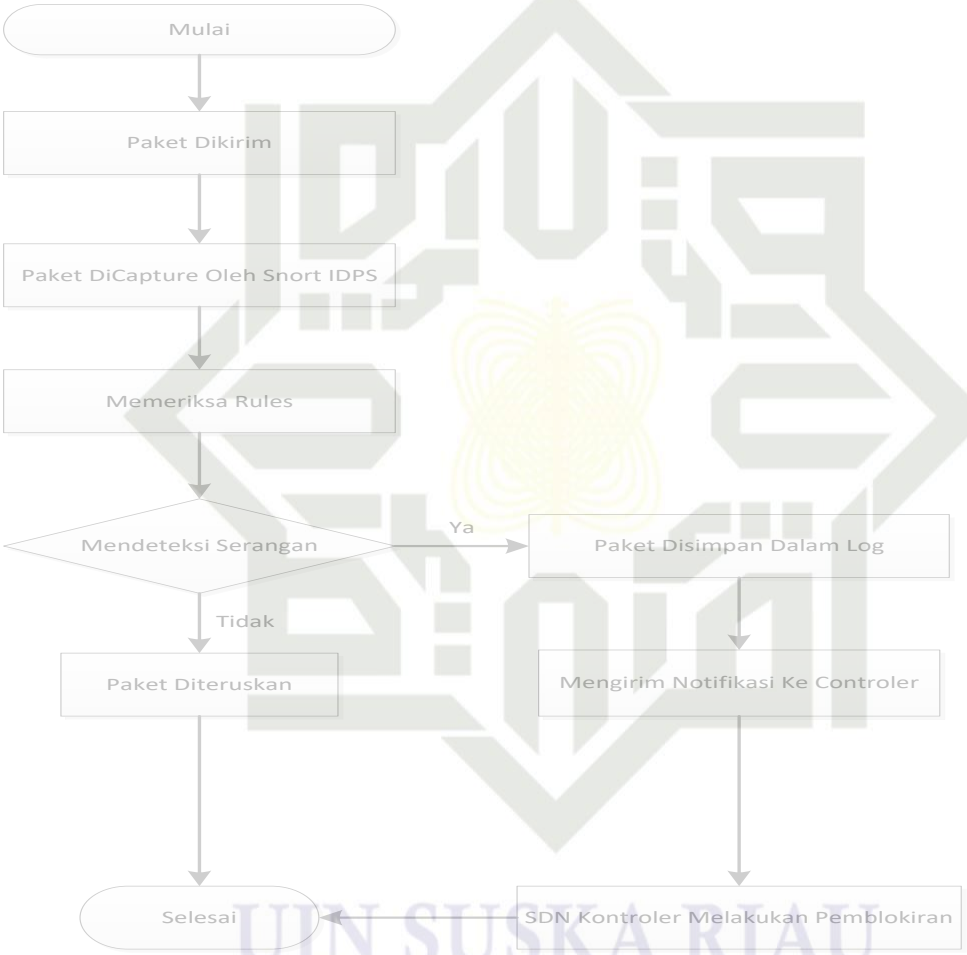
Gambar 3.4 Tampilan Rancangan Topologi Jaringan *Software Defined Network* (SDN)

Topologi jaringan *Software Defined Network* (SDN) dijalankan pada emulator *mininet* dan *controller* SDN menggunakan *Open Network Operating System* (ONOS). Perancangan topologi jaringan SDN dirancang berdasarkan topologi Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau dengan membuat sebuah program *file* .py (*python*) yang diberi nama *file* yaitu topologi.py pada *custom* topologi, kemudian akan dijalankan pada emulator *mininet* dan ONOS *controller*.

3.4.2 Rancangan *Intrusion Detection and Prevention System* (IDPS)

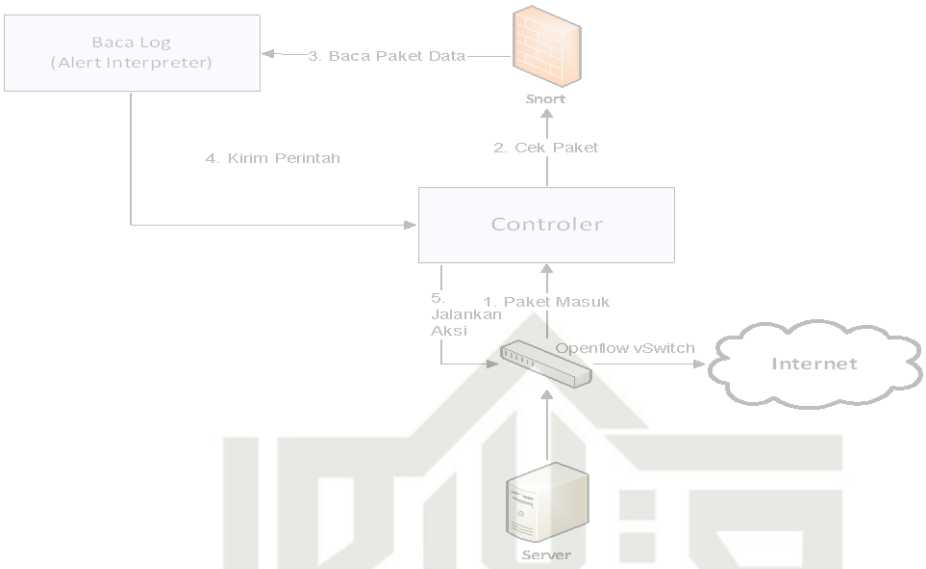
Pada umumnya, *Intrusion Detection and Prevention System* (IDPS) akan disimulsikan didalam jaringan *Software Defined Network* (SDN) bertujuan untuk memonitoring serta mendeteksi dan mencegah serangan secara *real* pada jaringan SDN. Aplikasi *opensource* yang memiliki IDPS didalamnya adalah *snort*. *Snort* akan dipasang pada kontroler SDN. Sebab, apabila semua paket data yang melewati kontroler SDN dapat

forward, kemudian *snort* akan membaca semua paket data yang masuk. Jika terdapat paket data yang berbahaya, maka *snort* mencatat *log* dari paket data tersebut. Kemudian, *snort* akan memberikan notifikasi ke *controller*, kemudian *controller* akan melakukan pencegahan (*blocking*) terhadap paket tersebut. Jika paket data tidak terdeteksi adanya bahaya, maka paket data akan diteruskan. Berikut adalah alur sistem digambarkan:



Gambar 3.5 Diagram Alur *Intrusion Detection and Prevention System* (IDPS)

Saat terjadi serangan didalam jaringan pemblokiran tidak terjadi secara otomatis karena *snort* hanya mencatat *log* serangan yang terjadi didalam jaringan. Namun, pemblokiran dapat dibuat menjadi otomatis dengan membuat sebuah program *alert interpreter* yang akan menerjemahkan *log* dari *snort*, kemudian *controller* akan melakukan pemblokiran. Alur program *alert interpreter* dalam pemblokiran adalah sebagai berikut :



Gambar 3.6 Cara Kerja *Intrusion Detection and Prevention System* (IDPS)

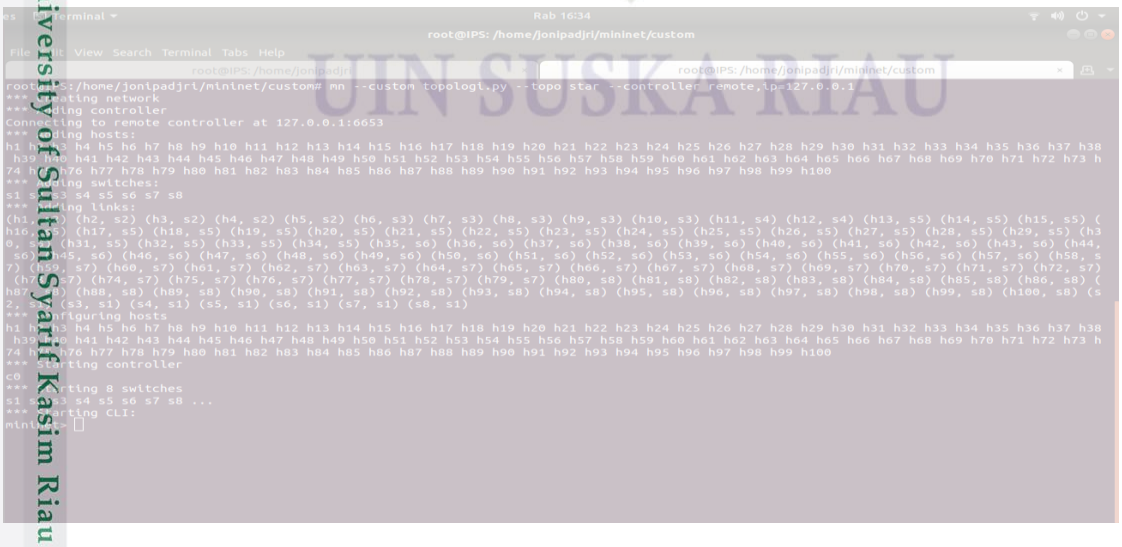
3.5 Pengujian Sistem

3.5.1 Pengujian Rancangan Jaringan *Software Defined Network* (SDN)

Pada penelitian ini penulis akan melakukan pengujian rancangan jaringan *Software Defined Network* (SDN) dilaksnakan di Pusat Teknologi Informasi dan Pangkalan Data (PTIPD). Berikut adalaha tahap-tahap pengujian jaringan SDN.

1. Untuk mengetahui apakah program yang sudah dibuat pada tahap sebelumnya dapat berjalan, maka dilakukan eksekusi program *file* ‘topologi.py’ pada *mininet* dan *Open Network Operating System* (ONOS) *controller* di terminal ubuntu. Eksekusi dilakukan dengan perintah:

mn –custom topologi.py --topo star –controller remote,ip=127.0.0.1



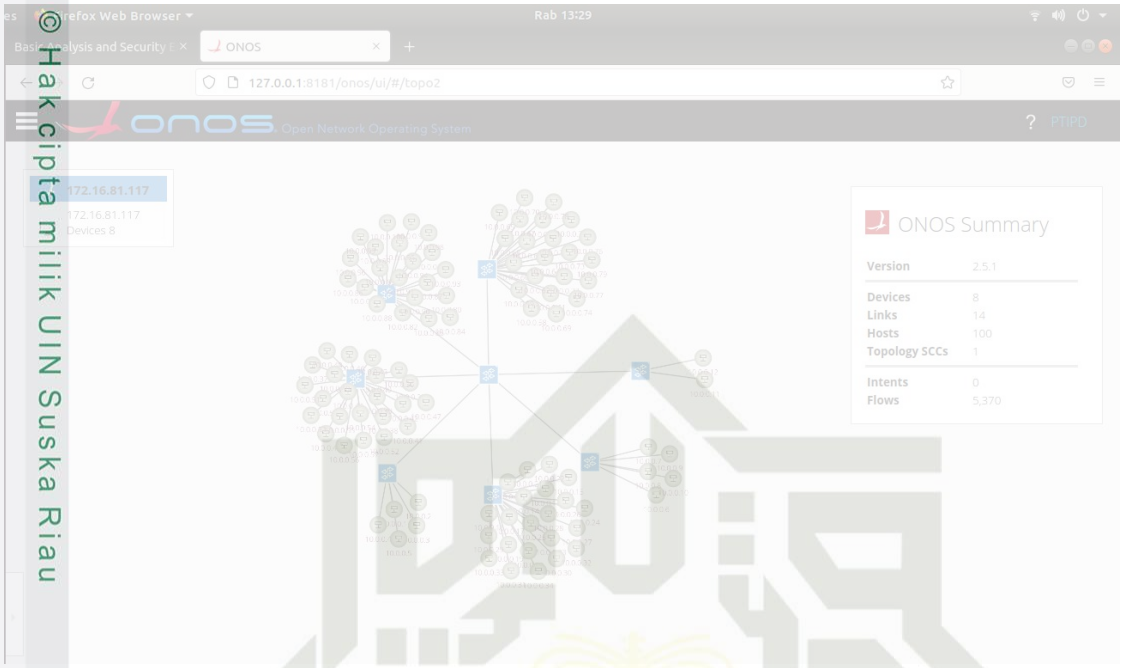
Gambar 3.7 Tampilan Eksekusi Program Di *Mininet*

b) Kemudian, untuk mengetahui apakah *mininet* terkoneksi dengan *Open Network Operating System* (ONOS) *controller*, maka dilakukan *test pingall* dan *test ping* antara *host* dan *host* di *mininet.*

Gambar 3.8 Tampilan *Test Pingall*

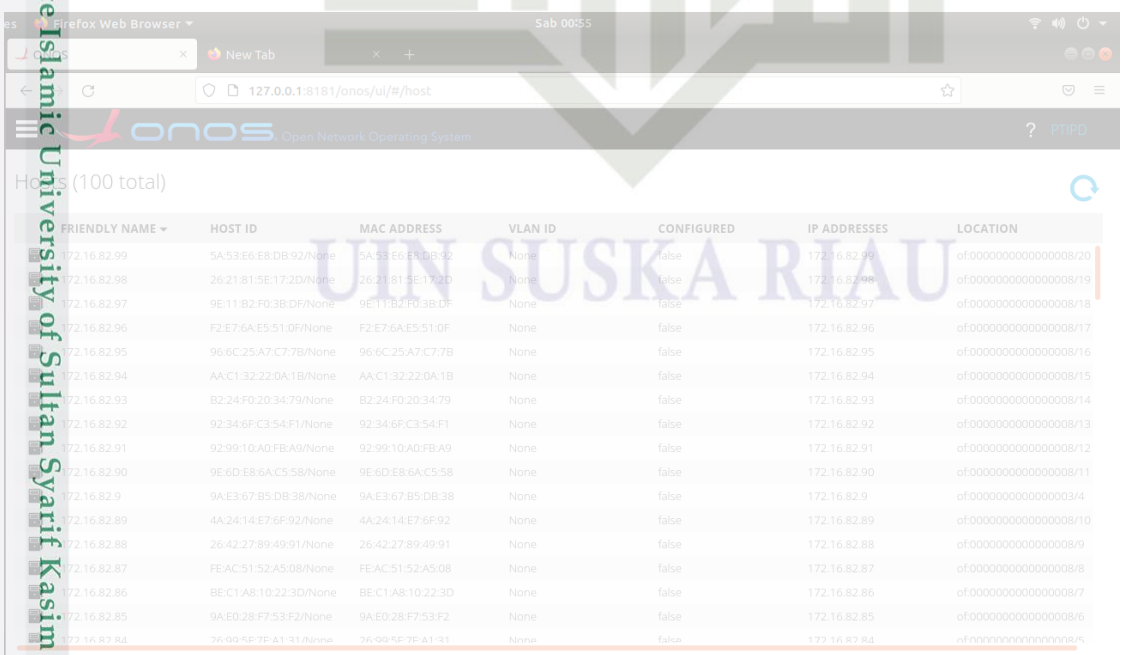
Gambar 3.9 Tampilan *Test Ping* antara *Host* dan *Host* Di *Mininet*

c) Untuk menampilkan rancangan topologi jaringan SDN berbasis *web Graphical User Interface* (GUI) pada ONOS *controller* dengan mengakses http://127.0.0.1:8181/onos/ui/index.html.



Gambar 3.10 Tampilan Topologi Jaringan Pada *Open Network Operating System* (ONOS)

d) Berikut ini adalah tampilan IP address *Open Network Operating System* (ONOS) kontroler berdasarkan *file* program "topologi.py" yang sudah di program sebelumnya dengan jumlah keseluruhan *host* sebanyak 100, ditampilkan dalam Gambar 3.11 adalah sebagai berikut.



Gambar 3.11 Tampilan IP Address *Open Network Operating System* (ONOS)

e) Untuk mengukur *Quality of Service* (QoS) rancangan jaringan yang sudah dibuat pada tahap sebelumnya yaitu dengan menjalankan perintah di terminal ubuntu "iperf3 –s" pada *server* dan perintah "iperf3 –c 172.16.81.117" pada user dan dihitung sebanyak 3 kali pengujian menggunakan aplikasi *Wireshark*.

Gambar 3.12 Tampilan Perintah *Iperf*3 Server

Gambar 3.13 Tampilan Perintah *Iperf*3 *User*

3.5.2 Pengujian serangan *Distributed Denial of Service* (DDoS)

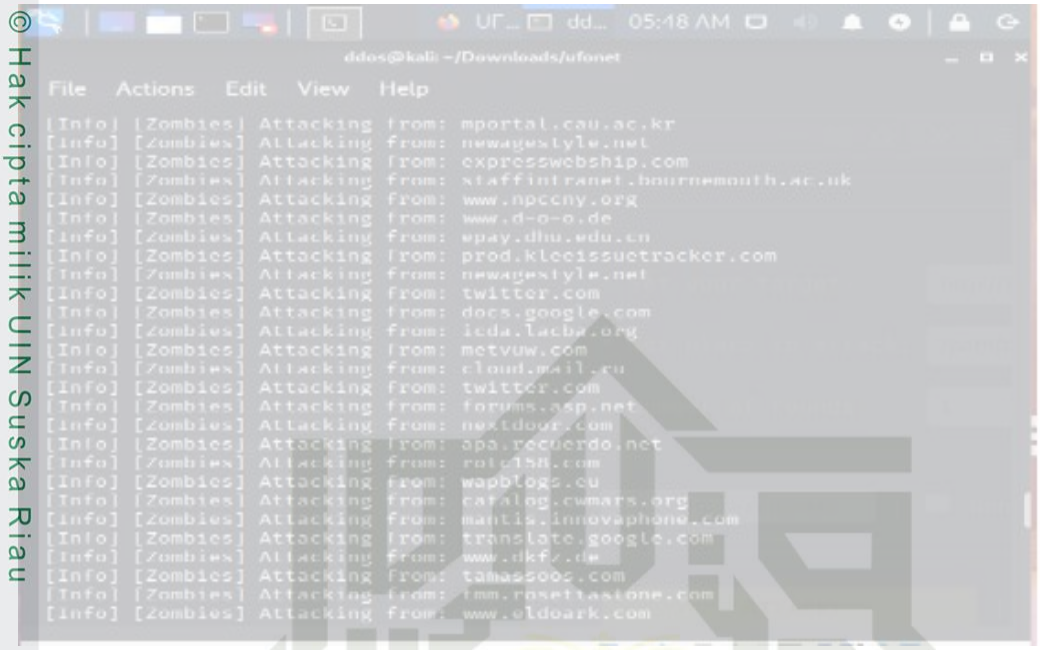
Pada penelitian ini penulis akan melakukan pengujian serangan *Distributed Denial of Service* (DDoS) yaitu *Botnet Attack* menggunakan aplikasi *ufonet.* Berikut adalah tahap-tahap pengujian serangan *botnet*.

1. Download *Zombie* atau *bot* pada terminal kalilinux.

Gambar 3.14 Tampilan *Download Bot*

2. Eksekusi serangan b*otnet*

Gambar 3.15 Tampilan Eksekusi Serangan *Botnet*



Gambar 3.16 Tampilan Proses Serangan *Botnet*

3.5.3 Pengujian *Intrusion Detection and Prevention System* (IDPS)

Pada penelitian ini penulis akan melakukan pengujian *Intrusion Detection and Prevention System* (IDPS) menggunakan snort, *IPTables*, dan *Basic Analysis Security Engine* (BASE). Berikut ini adalah tahap-tahap pengujian *Snort* IDPS.

1. Pengujian *snort* dengan membuat sebuah aturan (rules) bertujuan agar *snort* dapat dijalankan berdasarkan aturan (rules) yang dibuat.
   "alert tcp any any -> $HOME_NET any (msg:"DDoS"; GID:1; sid:100000001; rev:001; classtype:icmp-event;)".
2. Kemudian menjalankan *snort* pada terminal ubuntu dengan perintah adalah sebagai berikut.
   "sudo snort –A console –c /etc/snort/snort.conf –I wlp2s0"

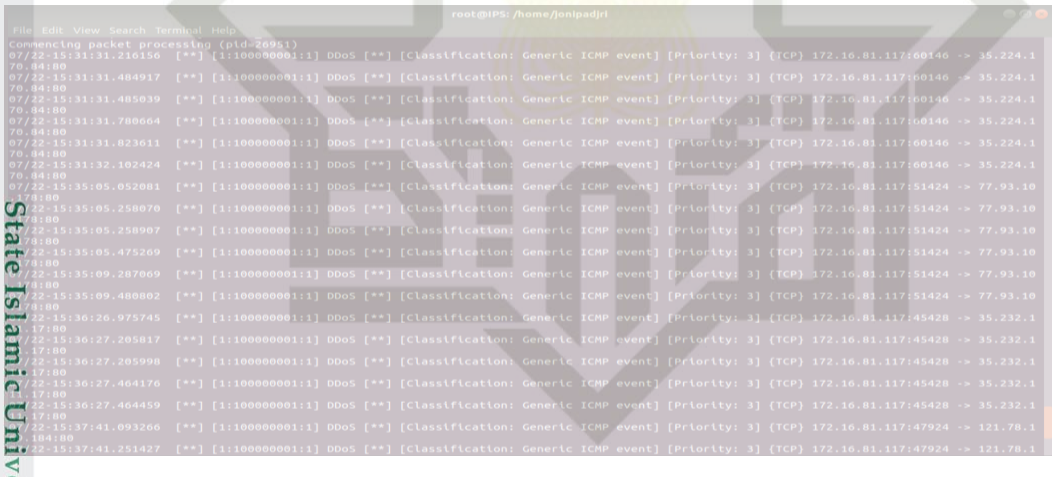
Gambar 3.17 Tampilan Perintah Menjalankan *Snort*

3. Selanjutnya menjalanakan perintah IP*Tables* pada terminal ubuntu. IP*Tables* merupakan sebuah *firewall* jaringan yang berfungsi sebagai *tools* guna melakukan filter (penyaringan) terhadap *trafic* atau lalulintas data dan memblokir serangan. Adapun perintah menjalankan IP*Tables* yaitu:

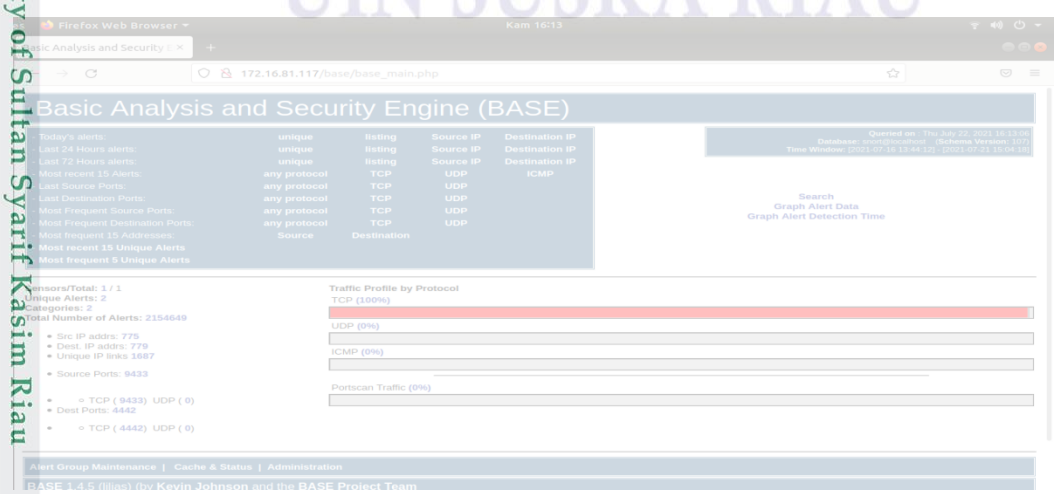
"iptables –A FORWARD –j NFQUEUE --queue-num 0"

Gambar 3.18 Tampilan Perintah Menjalankan IP*Tables*

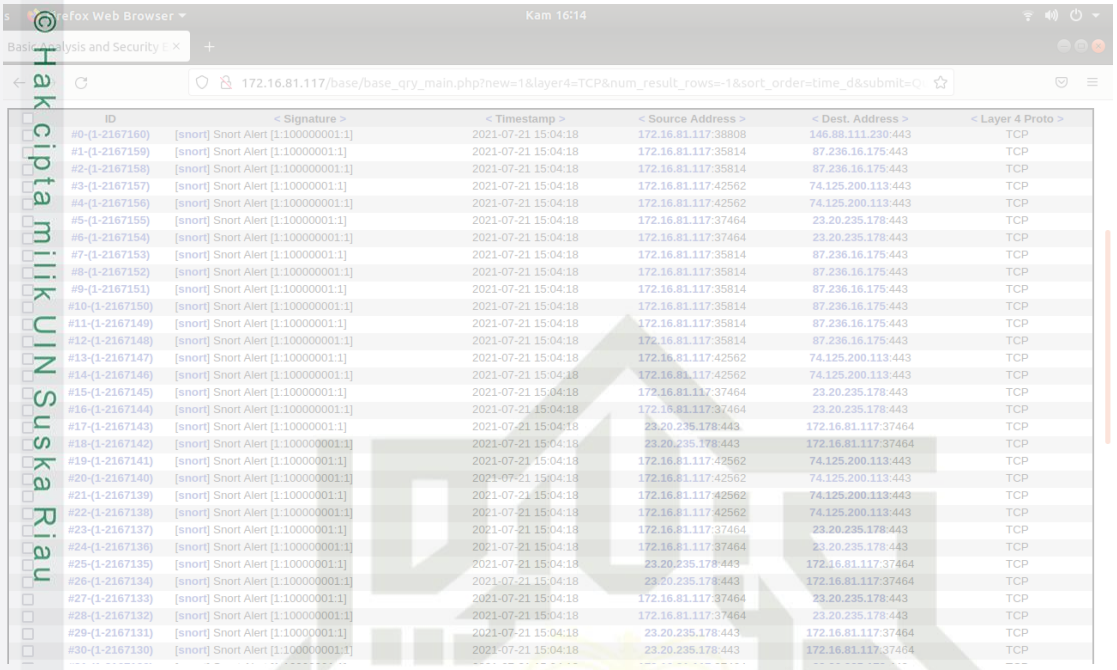
4. Kemudian *snort* akan memberikan pesan notifikasi diterminal dan *snort web Basic Analysis Security Engine* (BASE).



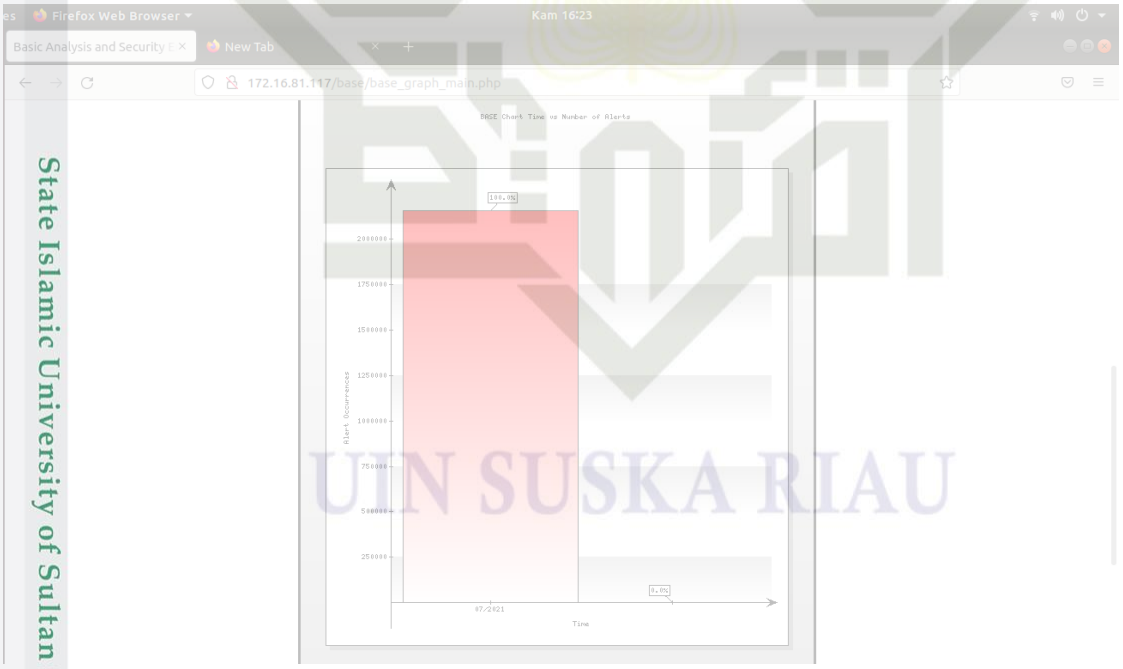
Gambar 3.19 Tampilan Notifikasi *Snort*



Gambar 3.20 Tampilan *Snort Web Basic Analysis Security Engine* (BASE)

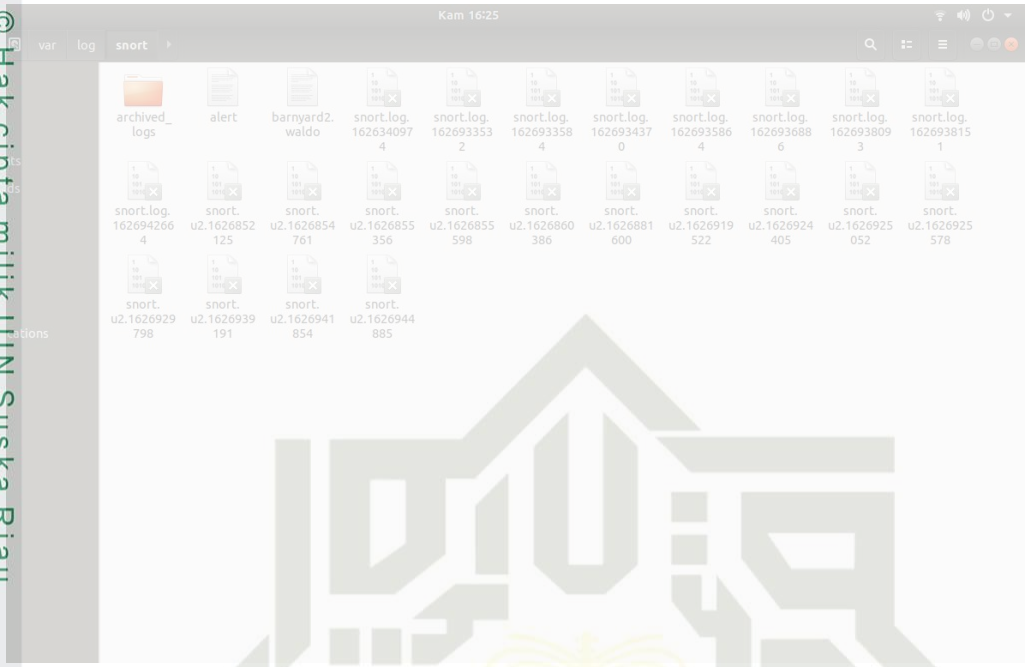


Gambar 3.21 Tampilan Keluaran *Snort* Web *Basic Analysis Security Engine* (BASE)



Gambar 3.22 Tampilan *Snort Web* Grafik *Basic Analysis Security Engine* (BASE)

5. kemudian, *snort* akan menyimpan *log* serangan di folder /var/log/snort/. Gambar 3.24 adalah tampilan *error* serangan *botnet* pada saat *snort* berjalan.



Gambar 3.23 Tampilan *Log Snort*

Gambar 3.24 Tampilan *Error* Proses Serangan *Botnet*

3.6 Parameter Pengujian

Quality of Service (QoS) mengacu pada kemampuan jaringan untuk mengetahui dan untuk mendapatkan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. Pada penelitian ini pengukuran *Quality of Service* (QoS) menggunakan aplikasi *wireshark.* Digunakan pada saat mengukur jaringan *Software Defined Network* dan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau, kemudian mengukur serangan *Distributed Denial of Service* (DDoS), dan mengukur *snort Intrusion Detection and Prevention System* (IDPS). Parameter *Quality of Service* (QoS) yang digunakan adalah.

1. *Throughput*

Throughput merupakan jumlah total kedatangan paket yang berhasil yang diamati pada tujuan selama jarak waktu tertentu dibagi oleh durasi jarak waktu tersebut. *Throughput* merupakan kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya *throughput* selalu dikaitkan dengan *bandwidth* karena *throughput* memang bisa disebut juga dengan *bandwidth* dalam kondisi yang sebenarnya. *Bandwidth* lebih bersifat *fix* sementara, sedangkan *throughput* sifatnya adalah dinamis tergantung trafik yang sedang terjadi. *Throughput* dapat dicari dengan persamaan sebagai berikut :

$$Throughput = \frac{\text{Jumlah Data Yang Dikirim(bytes)}}{\text{Waktu Pengiriman Data (s)}} \quad \text{.....(3.1)}$$

Tabel 3.1 Kategori *throughput*

Kategori Throughput	Throughput	Indek
Sangat Bagus	100 bps	4
Bagus	75 bps	3
Sedang	50 bps	2
Buruk	<25 bps	1

(Sumber: ETSI 1999 - 2006)

2. *Packet Loss*

Packet Loss adalah kegagalan transmisi paket IP mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinkan, diantaranya yaitu:

1. Terjadinya *overload* trafik didalamjaringan.
2. Tabrakan (*congestion*) dalam jaringan.

3. *Error* yang terjadi pada media fisik.
4. Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena *overflow* yang terjadi pada buffer.

$$PacketLoss = \frac{\text{Paket Dikirim} - \text{Paket Diterima}}{\text{Paket Dikirim}} x100\% \quad \text{.....(3.2)}$$

Tabel 3.2 Kategori *Packet Loss*

Kategori Packet Loss	Packet Loss	Indek
Sangat Bagus	0% - 2%	4
Bagus	3% - 14%	3
Sedang	15% - 24%	2
Buruk	>25%	1

(Sumber: ETSI 1999 - 2006)

3. *Delay*

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. *Delay* dapat dicari dengan persamaan sebagai berikut :

$$Delay = \frac{Waktu\ Total}{Jumlah\ Paket} \quad \text{.....(3.3)}$$

Tabel 3.3 Kategori *Delay*

Kategori Delay	Besar Delay	Indek
Sangat Bagus	<150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 ms s/d 450 ms	2
Buruk	>450 ms	1

(Sumber: ETSI 1999 - 2006)

BAB V
KESIMPULAN DAN SARAN

Kesimpulan dan saran yang dapat diambil dari Tugas Akhir (TA) yang berjudul Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection and Prevention System* (IDPS) (Studi Kasus: Pusat Teknologi Informasi Dan Pangkalan Data (PTIPD) UIN Suska Riau)

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan yaitu:

1. Hasil *throughput* jaringan *Software Defined Network* (SDN) lebih baik dengan nilai *throughput* 6.804 bps dibandingkan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau dengan nilai *throughput* 5.029 bps. Secara signifikan nilai *throughput* jaringan *Software Defined Network* (SDN) dan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau kategori "sangat bagus".
2. Hasil *delay* jaringan *Software Defined Network* (SDN) lebih baik dengan nilai *delay* 0,00111 ms dibandingkan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau dengan nilai *delay* 0,00115 ms. Secara signifikan nilai *delay* jaringan *Software Defined Network* (SDN) dan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau kategori "sangat bagus".
3. Hasil *packet loss* jaringan *Software Defined Network* (SDN) lebih baik dengan nilai *packet loss* 0,34% dibandingkan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau dengan nilai *packet loss* 0,89% ms. Secara signifikan nilai *packet loss* jaringan *Software Defined Network* (SDN) dan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau kategori "sangat bagus".
4. Berdasarkan hasil *Quality of Service* (QoS) dengan parameter *throughput*, *delay* dan *packet loss* jaringan *Software Defined Network* (SDN) lebih baik dibandingkan jaringan Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) UIN Suska Riau.

5. Berdasarkan hasil *Quality of Service* (QoS) jaringan *Software Defined Network* (SDN) dari pengujian serangan *botnet*, nilai *throughput* 8,743 bps kategori buruk, niali *delay* 0,0767 ms kategori bagus, dan nilai *packet loss* 29,43% kategori buruk.
6. Berdasarkan hasil *Quality of Service* (QoS) jaringan *Software Defined Network* (SDN) dari pengujian *snort* berbasis *Intrusion Detection and Prevention System* (IDPS) dan serangan *botnet.* Nilai *throughput* 414,67 bps kategori sangat bagus, nilai *delay* 0,00771 ms kategori sangat bagus, nilai *packet loss* 4,14% kategori bagus.
7. Berdasarkan hasil *Quality of Service* (QoS) snort berbasis *Intrusion Detection and Prevention System* (IDPS) mampu merubah kualitas jaringan *Software Defined Network* (SDN) menjadi lebih baik.

5.2 Saran

Adapun saran yang dapat diberikan untuk penelitian selanjutnya adalah:

1. Untuk perkembangan penelitian, disarankan menggunakan komputer dengan spesifikasi yang lebih tinggi.
2. Memberi masukan kepada administartor jaringan Pusat Teknologi Informasi Dan Pangkalan Data (PTIPD) UIN Suska Riau menggunakan sistem keamanan snort *Intrusion Detection and Prevention System* sebagai sistem keamanan jaringannya yang berbasis *opensource* sesuai yang dibutuhkan PTIPD.
3. Untuk penelitian selanjutnya diharapkan dilakukan dengan penerapan sistem keamanan jaringan secara *real* pada jaringan Pusat Teknologi Informasi Dan Pangkalan Data (PTIPD) UIN Suska Riau berbasis *Software Defined Network* (SDN) untuk mamanajemen jaringan dan keamanan jaringan secara otomatis dan terpusat.

DAFTAR PUSTAKA

[1] Mulyana, Eueng. 2015. "*Buku Komunitas SDN-RG*". Bandung : Gitbook.

[2] Mahesi, P. T., Dwi, F., & Sumadi, S. (2018). Analisis Dampak Serangan Distributed Denial of Service Pada Jaringan Openflow. *Seminar Nasional Teknologi Dan Rekayasa (SENTRA)*, 90–95

[3] Yasin, Alimuddin,. Mohidin, Ismail. (2018). *Dampak Serangan DDoS pada Software Based Openfow Switch di Perangkat HG553*. Jurnal Technopreneur (JTech), 72-74.

[4] Nugroho, M. A., & Suwastika, N. A. (2018). *Perancangan Intrusion Prevention System pada Jaringan Software Defined Networks*. Jumanji, 02(01), 1–16.

[5] Fierre. R. (2018). *Simulasi Pencegahan Serangan Denial Of Service (DoS) Pada Software-Defined Networking (SDN) Menggunakan Intrusion Prevention System (IPS) Dan Algoritma Genetika*. Fakultas Ilmu Komputer Dan Teknologi Informasi.

[6] Putu, P., Adi, K., Negara, R. M., Sanjoyo, D. D., Elektro, F. T., & Telkom, U. (2018). *Integrasi Intrusion Prevention System Dan Analisa Performansi Pada Software Defined Network Intrusion Prevention System Integration and Performance*, *5*(3), 5046–5053.

[7] Ummah, I. (2016). Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking. *Indonesian Journal on Computing (Indo-JC)*, *1*(1).

[8] "Binus Overview." [Online]. Available: https://socs.binus.ac.id/2018/12/10/software-defined-networking-sdn/. [Diakses 14 Juni 2021]

[9] "TechTarget Overview." [Online]. Available: https://searchnetworking.techtarget.com definition/software-defined-networking-SDN. [14 Juni 2021]

[10] "ONF Overview." [Online]. Available: https://opennetworking.org/onos/. [Diakses 14 Juni 2021]

[11] "Cyberthreat.id Overview." [Online]. Available: https://cyberthreat.id/read/5184/Yuk-Simak-Lagi-Beda-Serangan-DoS-dan-DDoS. [Diakses 14 Juni 2021]

[12] Putri, V. U., Cahyono, E. B., & Azhar, Y. (2020). Deteksi Botnet Pada Passive DNS Dengan Menggunakan Metode K Nearest Neighbor. *Jurnal Repositor*, *2*(12), 1631.

[13] Ursin, F., Yamin, M., & Surimi, L. (2017). Implementasi Security System Menggunakan Metode IDPS (Intrusion Detection and Prevention System) Dengan Layanan Realtime Notification. *SemanTIK*, *3*(2), 39–48.

[14] Suyuti Ma'sum, M., Azhar Irwansyah, M., & Priyanto, H. (2017). Analisis Perbandingan Sistem Keamanan Jaringan Menggunakan Snort dan Netfilter. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, *5*(1), 56–60.

[15] Adriant, M. F., & Mardianto, I. (2015). Implementasi *Wireshark* Untuk Penyadapan (*Sniffing*) Paket Data Jaringan. *Seminar Nasional Cendekiawan*, 224–228.

LAMPIRAN A

1. Ruangan Monitoring

Ruangan Monitoring terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 1. Terdapat 5 unit komputer dan 1 *switch*.

Gambar A.1 Ruangan Monitoring

2. Ruangan Aplikasi 1

Ruangan Aplikasi 1 terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 1. Terdapat 5 unit komputer dan 1 *switch*.

Gambar A.2 Ruangan Aplikasi 1

3. Ruangan Aplikasi 2

Ruangan Aplikasi 2 terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 1. Terdapat 2 unit komputer dan 1 *switch*.

Gambar A.3 Ruangan Aplikasi 2

4. Ruangan Laboratorium PTIPD A

Ruangan Laboratorium PTIPD A terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 3. Terdapat 22 unit komputer dan 1 *switch*.

Gambar A.4 Ruangan Laboratorium PTIPD A

6. Ruangan Laboratorium PTIPD B

Ruangan Laboratorium PTIPD B terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 3. Terdapat 23 unit komputer dan 1 *switch*.

Gambar A.5 Laboratorium PTIPD B

6. Ruangan Laboratorium PTIPD C

Ruangan Laboratorium PTIPD C terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 3. Terdapat 22 unit komputer dan 1 *switch*.

Gambar A.6 Laboratorium PTIPD C

. Ruangan Laboratorium PTIPD D

Ruangan Laboratorium PTIPD D terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 3. Terdapat 21 unit komputer dan 1 *switch*.

Gambar A.7 Ruangan Laboratorium PTIPD D

B. Ruangan Server

Ruangan Server terdapat di Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau pada lantai 3. Ruangan Server adalah ruangan yang terdapat semua perangkat jaringan inti (*switch* dan *router*) kampus UIN Suska Riau, yang mem*forward* setiap gedung atau bangunan di UIN Suska Riau.

Gambar A.8 Ruangan Server dan Perangkat Jaringan

9. Surat Keterangan Selesai Tugas Akhir

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PUSAT TEKNOLOGI INFORMASI DAN PANGKALAN DATA
مركزتكنولوخيا المعلومات وقواعد البيانات
CENTER OF INFORMATION TECHNOLOGY AND DATABASES
JL. HR. Soebrantas No. 155 KM. 18 Simpang Baru Panam Pekanbaru 28293 PO BOX 1004 HP 0811 7627 773
Website : www.ptipd.uin-suska.ac.id Email : ptipd@uin-suska.ac.id

SURAT KETERANGAN SELESAI TUGAS AKHIR
B-229/Un.04/UPT.II/PP.00.9/12/2020

Yang bertanda tangan di bawah ini menerangkan :

Nama	: Joni Padjri
Nim	: 11455101839
Jurusan	: Teknik Elektro
Semester	: XIII (Tiga Belas)

Telah selesai melaksanakan Penelitian Tugas Akhir di :

Perusahaan/Instansi	: Pusat Teknologi Informasi dan Pangkalan Data UIN Suska Riau
Alamat	: Jl. HR. Soebrantas No. 155 KM. 18 Simpang Baru Panam Pekanbaru
Bidang Kajian/Judul	: Perancangan Sistem Keamanan *Instrusion Prevention System* Laboratorium Fakultas Sains dan Teknologi UIN Suska Riau Berbasis Jaringan SDN
Pada Tanggal	: 07 Desember s.d. 17 Desember 2020

Pekanbaru, 18 Desember 2020
Kuasa Kepala PTIPD,

Dahyusman, S.Kom
NIP. 19800512 201101 1 001

SHOT ON POCO X3 NFC

0. Wawancara

WAWANCARA

Nama Narasumber : Indra Mulia Syafutra, S.T

Bekerja Di : Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau

Jabatan : Staff Pusat Teknologi Informasi dan Pangkalan Data (PTIPD)

Hasil Wawancara,

Mahasiswa : Topologi apa yang digunakan pada Pusat Teknologi Informasi dan Pangkalan Data (PTIPD), pak?

Narasumber : Disini menggunakan topologi star.

Mahasiswa : Di ruangan mana saja yang menggunakan jaringan komputer, pak? Dan berapa komputer yang terdapat pada ruangan tersebut?

Narasumber : Pada lantai 1 terdapat Ruangan monitoring ada 5 unit komputer dan 1 switch, Ruangan Aplikasi 1 ada 5 unit komputer dan 1 switch, Ruangan Aplikasi 2 ada 2 unit komputer dan 1 switch. Kemudian di lantai 3 terdapat 4 Ruangan laboratorium yaitu Ruangan Laboratorium PTIPD A ada 22 unit komputer dan 1 switch, Ruangan Laboratorium PTIPD B ada 23 unit komputer dan 1 switch, Ruangan Laboratorium PTIPD C ada 22 unit komputer dan 1 switch, dan Ruangan Laboratorium PTIPD D ada 21 unit komputer dan 1 switch.

Mahasiswa : Apakah cara konfigurasi jaringan pada perangkat jaringan disini masih konfigurasi satu per satu (manual), pak ?

Narasumber : iya, dengan cara manual.

Mahasiswa : Apakah pernah terjadi serangan atau gangguan pada server, pak?

Narasumber : Pernah, yaitu serangan *bomer* sejenis serangan DDoS.

Mahasiswa : Sistem kemanan apa yang digunakan saat serangan terjadi, pak?

Narasumber: Sistem kemanannya menggunakan *wazuh* untuk sistem monitoring atau analysis jaringan tetapi kekurangannya tidak dapat menyimpan log. Dan menggunakan *firewall* untuk mencegahnya menggunakan *fortigate* namun untuk kode lisensinya cukup mahal, kemungkinan akan mencari dan menggunakan sistem keamanan *opensource*.

Pekanbaru, 28 Juli 2021
Staff PTIPD UIN Suska Riau

Indra Mulia Syafutra, S.T.
NIP: 198105062011011005

Lt. Wawancara di What's App

LAMPIRAN B

List Program Perancangan Jaringan

```
"""Custom topology example

Two directly connected switches plus a host for each switch:

   host --- switch --- switch --- host

Adding the 'topos' dict with a key/value pair to generate our newly defined
topology enables one to pass in '--topo=mytopo' from the command line.
"""

from mininet.topo import Topo

class MyTopo( Topo ):
    "Simple topology example."

    def build( self ):
        "Create custom topo."

        # Add hosts
        h1 = self.addHost( 'h1', ip='172.16.82.1')
        h2 = self.addHost( 'h2', ip='172.16.82.2')
        h3 = self.addHost( 'h3', ip='172.16.82.3' )
        h4 = self.addHost( 'h4', ip='172.16.82.4')
        h5 = self.addHost( 'h5', ip='172.16.82.5')
        h6 = self.addHost( 'h6', ip='172.16.82.6')
        h7 = self.addHost( 'h7', ip='172.16.82.7')
        h8 = self.addHost( 'h8', ip='172.16.82.8')
        h9 = self.addHost( 'h9', ip='172.16.82.9')
        h10 = self.addHost( 'h10', ip='172.16.82.10')
        h11 = self.addHost( 'h11', ip='172.16.82.11' )
        h12 = self.addHost( 'h12', ip='172.16.82.12' )
        h13 = self.addHost( 'h13', ip='172.16.82.13' )
        h14 = self.addHost( 'h14', ip='172.16.82.14' )
        h15 = self.addHost( 'h15', ip='172.16.82.15' )
        h16 = self.addHost( 'h16', ip='172.16.82.16' )
        h17 = self.addHost( 'h17', ip='172.16.82.17' )
        h18 = self.addHost( 'h18', ip='172.16.82.18' )
        h19 = self.addHost( 'h19', ip='172.16.82.19' )
        h20 = self.addHost( 'h20', ip='172.16.82.20' )
        h21 = self.addHost( 'h21', ip='172.16.82.21' )
        h22 = self.addHost( 'h22', ip='172.16.82.22' )
        h23 = self.addHost( 'h24', ip='172.16.82.23' )
        h24 = self.addHost( 'h23', ip='172.16.8224.' )
        h25 = self.addHost( 'h25', ip='172.16.82.25' )
        h26 = self.addHost( 'h26', ip='172.16.82.26' )
        h27 = self.addHost( 'h27', ip='172.16.82.27' )
        h28 = self.addHost( 'h28', ip='172.16.82.28' )
        h29 = self.addHost( 'h29', ip='172.16.82.29' )
```

```
h56 = self.addHost( 'h56', ip='172.16.82.56' )
h57 = self.addHost( 'h57', ip='172.16.82.57' )
h58 = self.addHost( 'h58', ip='172.16.82.58' )
h59 = self.addHost( 'h59', ip='172.16.82.59' )
h60 = self.addHost( 'h60', ip='172.16.82.60' )
h61 = self.addHost( 'h61', ip='172.16.82.61' )
h62 = self.addHost( 'h62', ip='172.16.82.62' )
h63 = self.addHost( 'h63', ip='172.16.82.63')
h64 = self.addHost( 'h64', ip='172.16.82.64' )
h65 = self.addHost( 'h65', ip='172.16.82.65' )
h66 = self.addHost( 'h66', ip='172.16.82.66' )
h67 = self.addHost( 'h67', ip='172.16.82.67' )
h68 = self.addHost( 'h68', ip='172.16.82.68' )
h69 = self.addHost( 'h69', ip='172.16.82.69' )
h70 = self.addHost( 'h70', ip='172.16.82.70' )
h71 = self.addHost( 'h71', ip='172.16.82.71' )
h72 = self.addHost( 'h72', ip='172.16.82.72' )
h73 = self.addHost( 'h73', ip='172.16.82.73' )
h74 = self.addHost( 'h74', ip='172.16.82.74' )
h75 = self.addHost( 'h75', ip='172.16.82.75' )
h76 = self.addHost( 'h76', ip='172.16.82.76' )
h77 = self.addHost( 'h77', ip='172.16.82.77' )
h78 = self.addHost( 'h78', ip='172.16.82.78' )
h79 = self.addHost( 'h79', ip='172.16.82.79' )
h80 = self.addHost( 'h80', ip='172.16.82.80' )
h81 = self.addHost( 'h81', ip='172.16.82.81' )
h82 = self.addHost( 'h82', ip='172.16.82.82' )
h83 = self.addHost( 'h83', ip='172.16.82.83' )
h84 = self.addHost( 'h84', ip='172.16.82.84' )
h85 = self.addHost( 'h85', ip='172.16.82.85' )
h86 = self.addHost( 'h86', ip='172.16.82.86' )
h87 = self.addHost( 'h87', ip='172.16.82.87' )
h88 = self.addHost( 'h88', ip='172.16.82.88' )
h89 = self.addHost( 'h89', ip='172.16.82.89' )
h90 = self.addHost( 'h90', ip='172.16.82.90')
h91 = self.addHost( 'h91', ip='172.16.82.91' )
h92 = self.addHost( 'h92', ip='172.16.82.92')
h93 = self.addHost( 'h93', ip='172.16.82.93' )
h94 = self.addHost( 'h94', ip='172.16.82.94' )
h95 = self.addHost( 'h95', ip='172.16.82.95' )
h96 = self.addHost( 'h96', ip='172.16.82.96' )
h97 = self.addHost( 'h97', ip='172.16.82.97' )
h98 = self.addHost( 'h98', ip='172.16.82.98' )
h99 = self.addHost( 'h99', ip='172.16.82.99' )
h100 = self.addHost( 'h100', ip='172.16.82.100' )

# Add  switches
s1 = self.addSwitch( 's1' )
s2 = self.addSwitch( 's2' )
```

```
        s3 = self.addSwitch( 's3' )
        s4 = self.addSwitch( 's4' )
        s5 = self.addSwitch( 's5' )
        s6 = self.addSwitch( 's6' )
        s7 = self.addSwitch( 's7' )
        s8 = self.addSwitch( 's8' )

        # Add links host to switch
        self.addLink( h1, s2 )
        self.addLink( h2, s2 )
        self.addLink( h3, s2 )
        self.addLink( h4, s2 )
        self.addLink( h5, s2 )
        self.addLink( h6, s3 )
        self.addLink( h7, s3 )
        self.addLink( h8, s3 )
        self.addLink( h9, s3 )
        self.addLink( h10, s3 )
        self.addLink( h11, s4 )
        self.addLink( h12, s4 )
        self.addLink( h13, s5 )
        self.addLink( h14, s5 )
        self.addLink( h15, s5 )
        self.addLink( h16, s5 )
        self.addLink( h17, s5 )
        self.addLink( h18, s5 )
        self.addLink( h19, s5 )
        self.addLink( h20, s5 )
        self.addLink( h21, s5 )
        self.addLink( h22, s5 )
        self.addLink( h23, s5 )
        self.addLink( h24, s5 )
        self.addLink( h25, s5 )
        self.addLink( h26, s5 )
        self.addLink( h27, s5 )
        self.addLink( h28, s5 )
        self.addLink( h29, s5 )
        self.addLink( h30, s5 )
        self.addLink( h31, s5 )
        self.addLink( h32, s5 )
        self.addLink( h33, s5 )
        self.addLink( h34, s5 )
        self.addLink( h35, s6 )
        self.addLink( h36, s6 )
        self.addLink( h37, s6 )
        self.addLink( h38, s6 )
        self.addLink( h39, s6 )
        self.addLink( h40, s6 )
        self.addLink( h41, s6 )
        self.addLink( h42, s6 )
        self.addLink( h43, s6 )
        self.addLink( h44, s6 )
        self.addLink( h45, s6 )
        self.addLink( h46, s6 )
        self.addLink( h47, s6 )
        self.addLink( h48, s6 )
        self.addLink( h49, s6 )
        self.addLink( h50, s6 )
        self.addLink( h51, s6 )
        self.addLink( h52, s6 )
        self.addLink( h53, s6 )
        self.addLink( h54, s6 )
        self.addLink( h55, s6 )
        self.addLink( h56, s6 )
        self.addLink( h57, s6 )
        self.addLink( h58, s7 )
        self.addLink( h59, s7 )
        self.addLink( h60, s7 )
        self.addLink( h61, s7 )
        self.addLink( h62, s7 )
        self.addLink( h63, s7 )
        self.addLink( h64, s7 )
```

```
        self.addLink( h65, s7 )
        self.addLink( h66, s7 )
        self.addLink( h67, s7 )
        self.addLink( h68, s7 )
        self.addLink( h69, s7 )
        self.addLink( h70, s7 )
        self.addLink( h71, s7 )
        self.addLink( h72, s7 )
        self.addLink( h73, s7 )
        self.addLink( h74, s7 )
        self.addLink( h75, s7 )
        self.addLink( h76, s7 )
        self.addLink( h77, s7 )
        self.addLink( h78, s7 )
        self.addLink( h79, s7 )
        self.addLink( h80, s8 )
        self.addLink( h81, s8 )
        self.addLink( h82, s8 )
        self.addLink( h83, s8 )
        self.addLink( h84, s8 )
        self.addLink( h85, s8 )
        self.addLink( h86, s8 )
        self.addLink( h87, s8 )
        self.addLink( h88, s8 )
        self.addLink( h89, s8 )
        self.addLink( h90, s8 )
        self.addLink( h91, s8 )
        self.addLink( h92, s8 )
        self.addLink( h93, s8 )
        self.addLink( h94, s8 )
        self.addLink( h95, s8 )
        self.addLink( h96, s8 )
        self.addLink( h97, s8 )
        self.addLink( h98, s8 )
        self.addLink( h99, s8 )
        self.addLink( h100, s8 )

        # Add links switch to switch
        self.addLink( s2, s1 )
        self.addLink( s3, s1 )
        self.addLink( s4, s1 )
        self.addLink( s5, s1 )
        self.addLink( s6, s1 )
        self.addLink( s7, s1 )
        self.addLink( s8, s1 )
topos = { 'star': ( lambda: MyTopo() )}
```

LAMPIRAN C

1. **Instalasi *Mininet***

a. *Download* M*ininet*

git clone git://github.com/mininet/mininet."

```
padjri@SDN-IPS:~$ git clone git://github.com/mininet/mininet
Cloning into 'mininet'...
remote: Enumerating objects: 9752, done.
remote: Total 9752 (delta 0), reused 0 (delta 0), pack-reused 9752
Receiving objects: 100% (9752/9752), 3.03 MiB | 1.04 MiB/s, done.
Resolving deltas: 100% (6470/6470), done.
padjri@SDN-IPS:~$
```

Gambar C.1 *Download Mininet*

b. *Install Mininet*

"/mininet/util/install.sh –a."

```
padjri@SDN-IPS:~$ mininet/util/install.sh
Detected Linux distribution: Ubuntu 18.04 bionic amd64
sys.version_info(major=2, minor=7, micro=17, releaselevel='final', serial=0)
Detected Python (python) version 2
Installing all packages except for -eix (doxypy, ivs, nox-classic)...
Install Mininet-compatible kernel if necessary
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
```

Gambar C.2 *Install Mininet*

Membuat program topologi pada custom topologi dengan Bahasa pemograman berbasis *python*.

```
"""Custom topology example

Two directly connected switches plus a host for each
switch:

   host --- switch --- switch --- host

Adding the 'topos' dict with a key/value pair to generate
our newly defined

topology enables one to pass in '--topo=mytopo' from the
command line.

"""

from mininet.topo import Topo

class MyTopo( Topo ):

    "Simple topology example."

    def build( self ):

        "Create custom topo."

    # Add host
    h1 = self.addHost( 'h1' )
    h2 = self.addHost( 'h2' )
    h3 = self.addHost( 'h3' )
    h4 = self.addHost( 'h4' )
    h5 = self.addHost( 'h5' )
    h6 = self.addHost( 'h6' )
    h7 = self.addHost( 'h7' )
    h8 = self.addHost( 'h8' )
    h9 = self.addHost( 'h9' )
```

Gambar C.3 Program Topologi Custom

2. Instalasi *Open Network Opearting System* (ONOS) kontroler

Instalasi *Open Network Opearting System* (ONOS) dilakukan diterminal Ubuntu. Berikut ini adalah langkah-langkah install ONOS.

a. Penambahan user sdn

```
padjri@SDN-IPS:~$ sudo adduser sdn --system --group
Adding system user `sdn' (UID 123) ...
Adding new group `sdn' (GID 127) ...
Adding new user `sdn' (UID 123) with group `sdn' ...
Creating home directory `/home/sdn' ...
padjri@SDN-IPS:~$
```

Gambar C.4 Penambahan User sdn

Joni Padjri lahir pada tanggal 14 Juni 1995 merupakan anak pertama dari Zainuddin Caniago dan Yuliarni, yang beralamat Jl. HR.Soebrantas. Penulis menempuh pendidikan sekolah dasar SD 003 Baloi Center Batam lulus pada tahun 2007 kemudian melanjutkan pendidikan sekolah menengah pertama SMPN 20 Pekanbaru lulus pada tahun 2011. Kemudian melanjutkan pendidikan Sekolah Menengah Kejuruan Hasanah Pekanbaru pada tahun 2014. Penulis melanjutkan jenjang pendidikan di perguruan tinggi Universitas Islam Sultan Syarif Kasim Riau dengan mengambil Program Studi Teknik Elektro konsentrasi Komputer dengan Penelitian Tugas akhir yang berjudul "**Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection And Prevention System* (Studi Kasus: Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau)".**

No. HP : 0822 4651 5851
E-Mail : joni.padjri@students.uin-suska.ac.id

b. *Install Java* dan *Python*

```
padjri@SDN-IPS:~$ sudo apt install git zip curl unzip python-minimal openjdk-11-
jdk -y
[sudo] password for padjri:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-minimal is already the newest version (2.7.15~rc1-1).
unzip is already the newest version (6.0-21ubuntu1).
zip is already the newest version (3.0-11build1).
curl is already the newest version (7.58.0-2ubuntu3.12).
git is already the newest version (1:2.17.1-1ubuntu0.7).
openjdk-11-jdk is already the newest version (11.0.9.1+1-0ubuntu1~18.04).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

Gambar C.5 *Install Java* dan *Python*

Download *Open Network Opearting System* (ONOS)

```
keamanan@padjri: /opt
File  Edit  View  Search  Terminal  Help
keamanan@padjri:/opt$ sudo wget https://repo1.maven.org/maven2/org/onosproject/o
nos-releases/2.5.1/onos-2.5.1.tar.gz
[sudo] password for keamanan:
--2021-06-23 11:42:38--  https://repo1.maven.org/maven2/org/onosproject/onos-rel
eases/2.5.1/onos-2.5.1.tar.gz
Resolving repo1.maven.org (repo1.maven.org)... 151.101.52.209
Connecting to repo1.maven.org (repo1.maven.org)|151.101.52.209|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 381223798 (364M) [application/x-gzip]
Saving to: 'onos-2.5.1.tar.gz'

onos-2.5.1.tar.gz     2%[                    ]   7,33M   190KB/s    eta 49m 16s
```

Gambar C.6 *Download Open Network Opearting System* (ONOS)

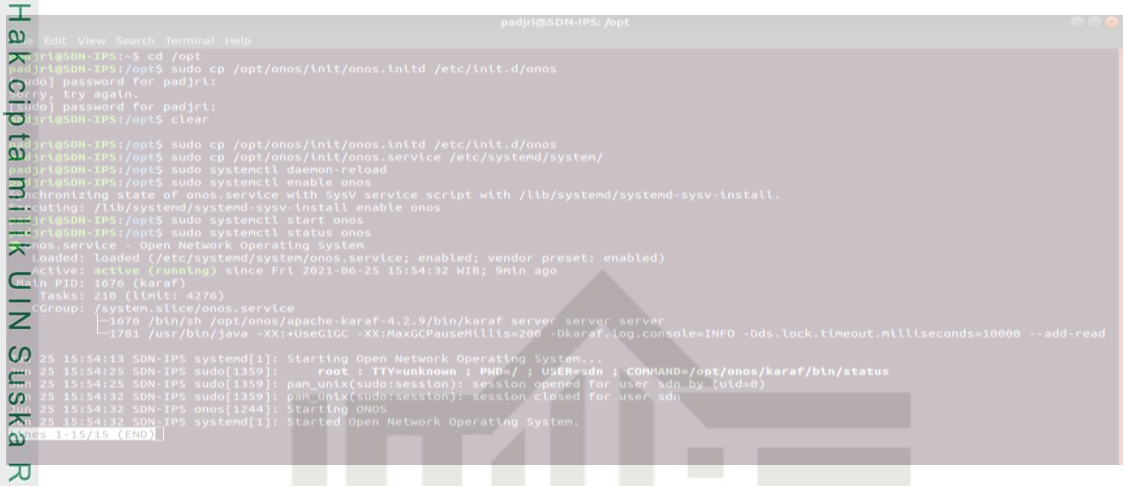
d. *Install Open Network Opearting System* (ONOS)

```
Terminal          Kam 16:57
padjri@SDN-IPS: /opt
File  Edit  View  Search  Terminal  Help
padjri@SDN-IPS:/opt$ sudo tar xzzf onos-2.5.1.tar.gz
padjri@SDN-IPS:/opt$ sudo mv onos-2.5.1 onos
padjri@SDN-IPS:/opt$ sudo chown -R sdn:sdn onos
padjri@SDN-IPS:/opt$ sudo -u sdn nano /opt/onos/options
Unable to create directory /home/padjri/.local/share/nano/: Permission denied
It is required for saving/loading search history or cursor positions.

Press Enter to continue
```

Gambar C.7 *Install Open Network Opearting System* (ONOS)

e. Start *Open Network Opearting System* (ONOS)



Gambar C.8 *Start Open Network Opearting System* (ONOS)

3. Instalasi *Snort* Data *Acquisition* (DaQ)

a. *Install packet Snort*

Gambar C.9 Install *Packet Snort*

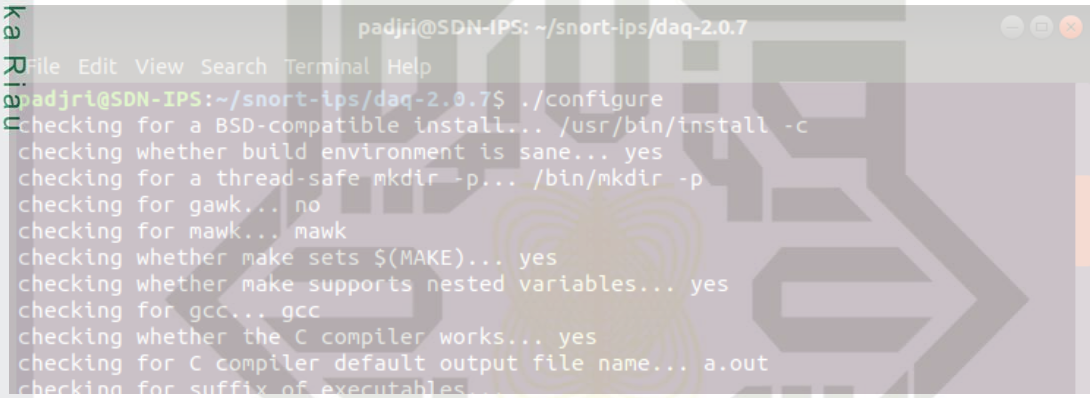
b. *Download* Data *Acquisition* (DaQ)

Gambar C.10 *Download* Data *Acquisition* (DaQ)

c. *Extract file* Data *Acquisition* (DaQ)

Gambar C.11 *Extract File*

d. Konfigurasi Data *Acquisition* (DaQ)



Gambar C.12 *Configure* Data *Acquisition* (DaQ)

e. *Install* Data *Acquisition* (DaQ)

Gambar C.13 *Install* Data *Acquisition* (DaQ)

4. **Instalasi *Snort***

a. *Download Snort*

Gambar C.14 *Download Snort*

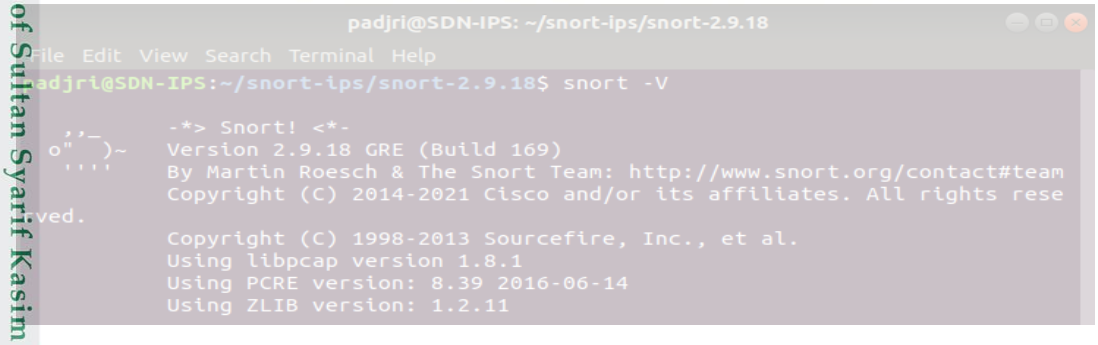
b. *Extract file snort*

Gambar C.15 *Extract File*

c. *Configure snort* dan *install*

Gambar C.16 konfigurasi dan *Install*

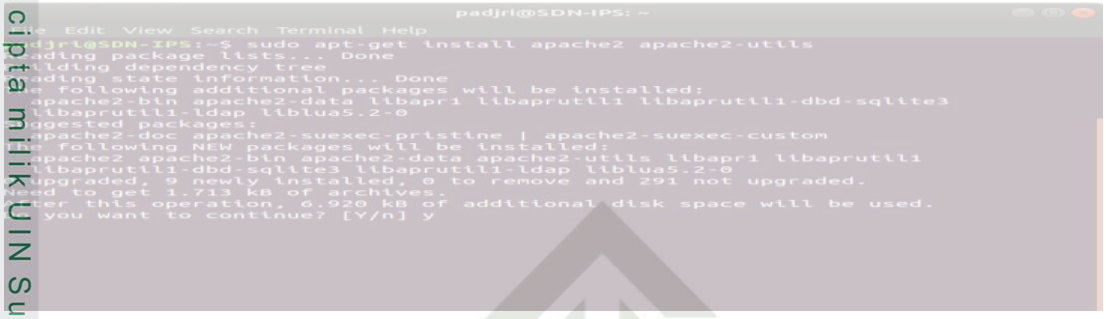
c. Cek versi *snort*



Gambar C.17 Cek Versi *Snort*

3. ***Apache* HTTP *Server***

a. Apt-get install *apache*2 *apache*2-utils



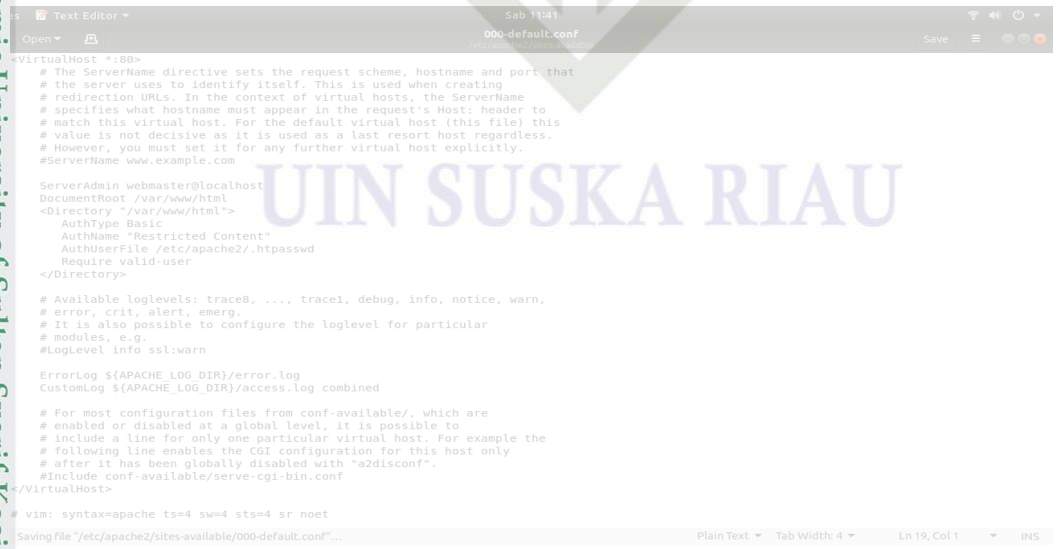
Gambar C.18 *Install Apache*2

b. Membuat *web* sederhana di index.html

Gambar C.19 *Index* HTML

c. Konfigurasi *apache*2

Gambar C.20 Perintah Konfigurasi *Apache*2 DiTerminal



Gambar C.21 Konfigurasi *Apache*2

d. Membuat *Username* dan *Password Server*

```
padjri@IPS:~$ sudo htpasswd -c /etc/apache2/.htpasswd PTIPD
password:
ype new password:
ng password for user PTIPD
padjri@IPS:~$
```

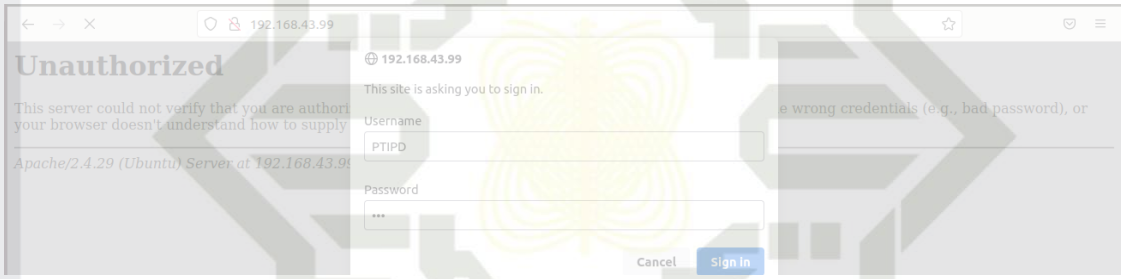
Gambar C.22 *Username* Dan *Password*

e. *Restart Apache2*

```
padjri@IPS:~$ sudo /etc/init.d/apache2 restart
 ] Restarting apache2 (via systemctl): apache2.service.
padjri@IPS:~$
```

Gambar C.23 *Restart Apache2*

f. Hasil Install dan Konfigurasi *apache2*



Gambar C.24 *Login Web Server*

Gambar C.25 *Web Server*

Joni Padjri lahir pada tanggal 14 Juni 1995 merupakan anak pertama dari Zainuddin Caniago dan Yuliarni, yang beralamat Jl. HR.Soebrantas. Penulis menempuh pendidikan sekolah dasar SD 003 Baloi Center Batam lulus pada tahun 2007 kemudian melanjutkan pendidikan sekolah menengah pertama SMPN 20 Pekanbaru lulus pada tahun 2011. Kemudian melanjutkan pendidikan Sekolah Menengah Kejuruan Hasanah Pekanbaru pada tahun 2014. Penulis melanjutkan jenjang pendidikan di perguruan tinggi Universitas Islam Sultan Syarif Kasim Riau dengan mengambil Program Studi Teknik Elektro konsentrasi Komputer dengan Penelitian Tugas akhir yang berjudul "**Perancangan Sistem Keamanan Jaringan Berbasis *Software Defined Network* (SDN) Menggunakan *Intrusion Detection And Prevention System* (Studi Kasus: Pusat Teknologi dan Informasi Pangkalan Data (PTIPD) UIN Suska Riau)".**

No. HP : 0822 4651 5851
E-Mail : joni.padjri@students.uin-suska.ac.id